FACE RECOGNITION - ALGORITHMIC APPROACH FOR LARGE DATASETS AND 3D BASED POINT CLOUDS

Ahmed ElSayed

Under the Supervision of:

Dr. Ausif Mahmood

Dr. Tarek Sobh

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIRMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOHPY IN COMPUTER SCIENCE

AND ENGINEERING

THE SCHOOL OF ENGINEERING

UNIVERSITY OF BRIDGEPORT

CONNECTICUT

April, 2017

FACE RECOGNITION - ALGORITHMIC APPROACH FOR LARGE DATASETS AND 3D BASED POINT CLOUDS

Ahmed ElSayed

Under the Supervision of: Dr. Ausif Mahmood, Dr. Tarek Sobh

Approvals

Committee Members

Name	**Signature**	**Date**
Dr. Ausif Mahmood	Ausif Mahmood	7-7-2017
Dr. Tarek Sobh	T. Sobh	7-7-2017
Dr. Jeongkyu Lee		7-6-2017
Dr. Prabir Patra	P. Patra	7-6-2017
Dr. Terrance Boult		7/7/17
Ph.D. Program Coordinator		
Dr. Khaled M. Elleithy		7/7/17
Chairman, Computer Science and Engineering Department		
Dr. Ausif Mahmood	Ausif Mahmood	7-7-2017
Dean, School of Engineering		
Dr. Tarek M. Sobh	T. Sobh	7-7-2017

FACE RECOGNITION - ALGORITHMIC APPROACH FOR LARGE DATASETS AND 3D BASED POINT CLOUDS

FACE RECOGNITION - ALGORITHMIC APPROACH FOR LARGE DATASETS AND 3D BASED POINT CLOUDS

ABSTRACT

This work proposes solutions for two different scenarios in face recognition and verification. The first scenario involves large scale unconstrained unsupervised face recognition. The proposed system for this scenario is a complete face recognition framework. The proposed system first studies the performance of unsupervised face recognition for frontalized captured faces in the wild under the effect of a single image super-resolution algorithm. The system also introduces new high dimensional features based on LBP and SURF that perform better than the state-of-the-art features for unconstrained unsupervised face recognition. To solve the large scale recognition process, a new algorithm has been designed to manipulate face images in the dataset. This new algorithm represents all training face images as a fully connected graph. The algorithm then divides the fully connected graph into simpler sub-graphs to enhance the overall recognition rate. The sub-graphs are generated dynamically, and a comparison between different sub-graph selection techniques including minimizing edge weight sums, random selection, and maximizing sum of edge weights inside the sub-graph is provided. Results show that the optimized hierarchical dynamic technique developed with sub-graphs selection increases the recognition rate in large benchmark image dataset by more than 40% for rank 1 recognition rate compared to the original single large graph method. The approach developed in this research is tested on different datasets, especially if the number of images per person in the

training data is low. Furthermore, in order to improve rank 1 recognition rates and to reduce the computation time of the recognition process, a new technique that combines the hierarchical face recognition algorithm and a deep learning neural network using Siamese structure for face verification is proposed.

The second part of this work addresses the usage of neural generative models for 3D faces with an application in face recognition when 3D datasets are utilized separately without the existence of texture information scenarios. An improved technique is developed to construct new representations for point clouds containing 3D information. The technique employs a regression neural network model trained using Levenberg-Marquardt (LM) algorithm. One of the advantages of this new representation is the significant reduction in storage space required for point clouds due to the utilization of a regression model for depth map regeneration. Moreover, the trained neural models can be used to generate a super-resolution version of the original 3D point clouds. The proposed regression representation is also used with a deep Siamese neural system to implement a complete depth-based neural face recognition and verification framework. The results indicate that the proposed system provides highly accurate and efficient face recognition results with 3D information only without texture information.

To my lovely family, my parents, my wife and my sons. I couldn't have done this without your support and encouragement. Thank you.

ACKNOWLEDGEMENTS

This work would not have been possible without the support of my advisors Dr. Ausif Mahmood and Dr. Tarek Sobh. Throughout the course of this dissertation, Dr. Mahmood and Dr. Sobh provided invaluable guidance, and worked tirelessly to help me craft this dissertation through several stages. Their encouragement and support were vital to the completion of this dissertation.

Also I would like extend my thanks to Dr. Elif Kongar for her help, support and valuable advises.

Moreover, would like to express my thanks to Dr. Khaled Elleithy for his support, advice, and encouragement.

I also would like to acknowledge the valuable contributions of the committee members, Dr. Jeongkyu Lee, Dr. Prabir Patra and Dr. Terrance Boult. I am grateful for their valuable feedback and guidance.

Finaly, I would like to thank all the members of the School of Engineering for their support, who made my studies at the University of Bridgeport a wonderful and exciting experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Who are you? A question that you ask someone you do not know or recognize. From the 1960's to the 1980's, computer scientists tried to develop automated algorithms to answer this question, and to provide the computer with the ability to identify a person based on a facial image. The problem became even more complex in the last two decades. One of the challenges involves the orientation and resolution of the captured face image, and the effect of these two factors on the recognition process. Another challenge stems from the extensive size of the search space caused by the datasets which are capable of storing several millions of images. Adding to this, the number of images available for each person is also another factor that needs to be taken into account. If the stored images in the dataset include multiple images per person, the task becomes relatively simple. However, the quality of the recognition algorithms gains significant importance if one single image per person is available in the dataset. The final consideration involves the availability of 3D information. That is, if we have only the depth information, i.e., points cloud, for faces for large numbers of individuals, how can this information be stored, and how can the stored data be regenerated and utilized for recognition purposes?

1.1 Research Problem and Motivations

This section provides detailed explanation regarding the problems addressed in the dissertation along with the motivation behind the proposed solutions. There are three

major problems covered in this research: 1) Unsupervised face recognition in the wild using high dimensional features under super-resolution and 3D alignment effect, 2) Hierarchical sub-graph selection (HSGS) algorithm with application for unsupervised face recognition, and 3) Face recognition for 3D data using neural generative models for features extraction and neural system for the verification process.

1.1.1 Effect of Super Resolution and 3D alignment on high dimensional features for unsupervised face recognition in the wild

Face recognition algorithms mostly utilize query faces captured from uncontrolled, in the wild, environments. The quality of these facial images are affected by various internal factors such as the quality of sensors used in outdoor cameras as well as external ones, such as the quality and direction of light. These factors adversely affect the overall quality of the captured images often causing blurring and/or low resolution. Super resolution algorithms are very effective in improving the resolution of these degraded images, more so if the captured face is small requiring scaling up. With this motivation, this research aims at demonstrating the effect of one of the state-of-the-art image super resolution algorithms on the labeled faces in the wild (lfw) dataset. In this regard, several cases are analyzed to demonstrate the effectiveness of the super-resolution algorithm. Each case is then investigated independently comparing the order of applying it before or after the 3D face alignment step. Following this, resulting images are tested on a closed set face recognition protocol using unsupervised algorithms with high dimensional extracted features. The inclusion of super resolution resulted an improvement in the recognition rate compared to unsupervised algorithm results reported in the literature.

1.1.2 Unsupervised sub-graph selection for large scale face recognition

One of the limitations of the existing face recognition algorithms is that the recognition rate significantly decreases with growing dataset sizes. In order to eliminate this shortcoming, this work presents a new training dataset partitioning methodology to improve face recognition for large datasets. This methodology is then applied to a common unsupervised recognition algorithm (Eigenface). Since the algorithm is mainly a data manipulation technique, it can also be used with other unsupervised algorithms or features, such as ICA, LBP, and SURF. The partitioning algorithm represents the training face images as a fully connected graph. This graph is then divided into simpler sub-graphs with hierarchical structure to enhance the overall recognition rate. The sub-graphs are generated dynamically, and a comparison between different sub-graph selection techniques including minimizing edge weight sums, random selection, and maximizing sum of edge weights inside the sub-graph are provided. Furthermore, in order to improve recognition rate, a highly accurate face verification system has been applied on the top 50 results of the hierarchy to improve rank 1 results. This corresponds to a verification rate with false acceptance rate (FAR) of 1%.

1.1.3 Neural generative model for 3D data with application in 3D face recognition and verification

3D face verification systems are complex and charged with difficult tasks. This is primarily caused by the variations in the resolutions of the 3D point clouds resulting from different cameras. Previous studies aim at solving this problem using 3D registration techniques. Out of these proposed techniques, detecting points of correspondence is proven to be efficient given that the data belongs to the same individual. However, if the data belongs to different persons, the registration algorithms

can convert the 3D point cloud of one person to the other destroying the distinguishing features between the two point clouds. Another issue relates to the storage size of the point clouds. That is, if the captured depth image contains around 50 thousand points in the cloud for a single pose for one individual, then the storage size of the entire dataset will be in order of giga if not tera bytes. With these motivations, this work introduces a new technique for 3D points cloud generation model using a neural system to handle the differences caused by heterogeneous depth cameras, and to generate a new face canonical compact representation. The proposed system reduces the stored 3D dataset size, and if required, provides an accurate dataset regeneration. Furthermore, the system generates neural models for all gallery point clouds and stores these models to represent the faces in the recognition or verification process. For the probe cloud to be verified, a new model will be generated specifically for that particular cloud and will be matched against pre-stored gallery model presentations to identify the query cloud. This work also introduces the utilization of Siames deep neural network in 3D face verification using generated model representation as a raw data for the deep network, and shows that the accuracy of the trained network outperforms all published results on Bosphorus dataset.

1.2 Potential Contributions of the Proposed Research

This work distinguishes itself from its counterparts and contributes to the related literature in two ways. First, for the 2D Face recognition system, the results show that the optimized hierarchical dynamic technique developed with sub-graphs selection increases the recognition rate in the benchmark image dataset by more than 40% for rank 1 recognition rate compared to the original single large graph method. The approach developed in this research has been tested on one of the popular unsupervised face recognition techniques (PCA), even though it can also be applied to

other unsupervised techniques including KPCA, ICA, LBP, and SURF. In addition, to ensure its functionality, the approach will utilize different datasets, especially if the number of images per person in the training data is low, viz., about one image per person. The second major contribution of this research is combining the hierarchical face recognition algorithm and a deep learning with Siamese neural network for face verification to improve rank 1 recognition rates while reducing the computational time of the recognition algorithm. The main contributions of the first part are listed in the following.

(1) Introducing a new unsupervised grouping technique for large training datasets,

(2) Applying different grouping criteria in the proposed method,

(3) Demonstrating the efficiency of the proposed method by providing a comparative study using multiple databases,

(4) Combining face recognition with face verification algorithms to improve final recognition rate and execution time,

(5) Merging the proposed recognition system with the state of the art techniques for 3D face alignment and super-resolution algorithm to increase the robustness of the proposed system,

(6) Introducing new multi-scales high dimensional features for unsupervised face recognition,

(7) Studying the effect of super-resolution and 3D alignment modules on the proposed high dimensional features.

For the second part, the mean square error (MSE) for training neural regression model for 3D face representation came out to be in order of 0.0002, implying that

the proposed neural generative model is able to accurately describe the provided data. Furthermore, the proposed 3D face representation is used with deep neural system based on Siamese architecture to implement a complete neural face recognition and verification for 3D data. The main contributions of the proposed 3D neural recognition and verification system are listed in as follows.

(1) Designing neural generative model for representation and reconstruction of 3D faces,

(2) Significant reduction in the storage space used for the 3D point clouds, by replacing the stored point clouds with the generated neural model representations,

(3) Using the generated presentation from the 3D regression models of gallery set for recognition and verification against generated model representation for probe points cloud,

(4) Combining generated face model representation with Siamese network to generate a comprehensive highly accurate framework for 3D texture free face recognition and verification.

The structure and chapters organization of the dissertation are shown in Figure 1.1.

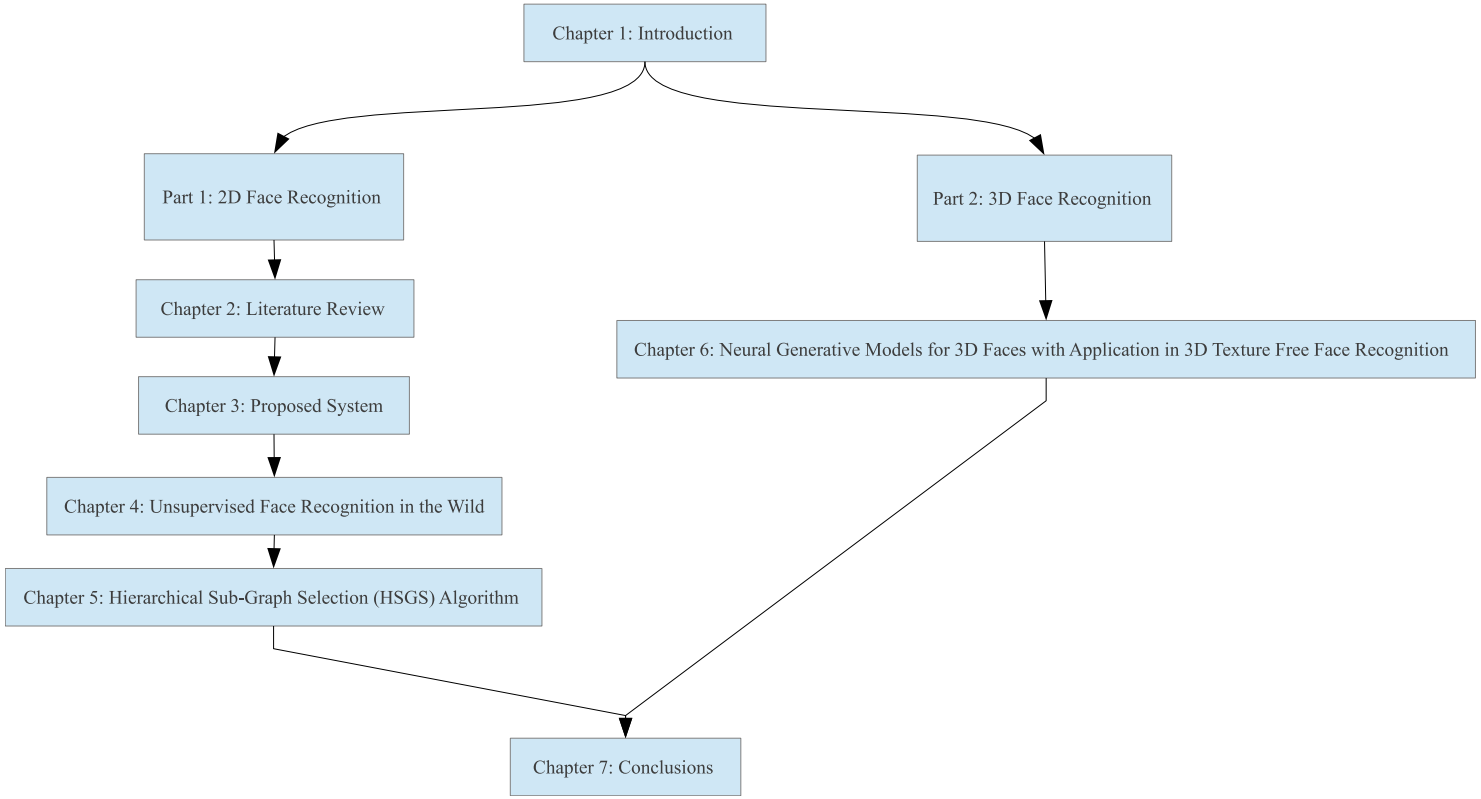


Figure 1.1: Dissertation organizational structure and chapters.

CHAPTER 2: LITERATURE SURVEY ON 2D FACE RECOGNITION

2.1 Face Recognition Techniques

Identity detection is an important problem in the fields of security and intelligence. Face recognition is one of the computer vision fields that is charged with this task. In this regard, several face recognition algorithms have been proposed and developed in the last decades including Direct Correlation, Principal Component Analysis (PCA) [1, 2, 3], Linear Discriminant Analysis (LDA) [4, 5], Independent Component Analysis (ICA) [2, 6, 7], Kernel methods (i.e KPCA and Support Vector Machine (SVM)) [8, 9], in addition to other high dimensional features methods such as Local Binary Pattern (LBP) [10, 11], Scale Invariant Feature Transform (SIFT) [12, 13] ... etc. Figure 2.1, Figure 2.2 [11] and Figure 2.3 [12] show examples of these features (a detailed explanation of these features will be discussed in Chapter 4). Some of these methods are supervised (e.g., LDA and SVM) where the given data is divided into training, testing and validation datasets. The training dataset is then divided into labeled groups (i.e., classes) with each class containing images of one person. The other methods for face recognition are unsupervised and use extracted features from faces (e.g., PCA, KPCA, ICA, LBP and SIFT) where the given data is separated into training and testing datasets. In this case, the training dataset is unlabeled and the algorithm handles the class separation process. Various research and experiments

have proven that simple recognition algorithms like PCA, ICA and SIFT produce good recognition rates with small sized (typically less than 100 images) datasets with accurately clipped face images. However, when the number of images in the dataset slightly exceeds hundred, the recognition accuracy of these algorithms reduces significantly.

One approach to handle this problem is to use indexing [14]. However, indexing is only applicable to specific features and techniques, in addition to being very sensitive to image registration, normalization, orientation and features calculation. The hierarchical recognition and partitioning technique can be used to improve the recognition rate when large datasets are required. This technique divides the given training dataset into smaller subgroups. This way the input image is compared with the stored images located in relatively smaller sets. The best matches are then selected and fed into the groups that are in the subsequent levels until a single small group remains. Finally, the best result from this final group determines a match or mismatch. Such hierarchical grouping principle on the training dataset has been used in [15, 16, 17]. These studies utilize a supervised grouping where the training dataset is divided according to the image class. In other words, the images of the same person are grouped together resulting in a clear separation between groups. One major drawback of this approach is that the group size is reduced to one when multiple images of the same person are not available. Furthermore, the number of groups increases significantly as the number of different individuals increase, especially when there are limited number of images for each person in the database.

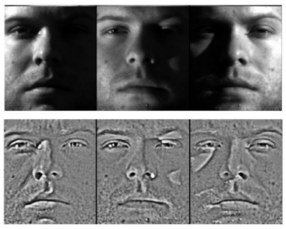
Another approach that has been used on large scale datasets is face verification. Face verification is different from face recognition in various aspects. Face recognition can be considered as one to many (1:N) relationship from features to classes space since it maps the features of a single image into the identity of different persons. On

Figure 2.1: Example bases generated using PCA (Eigenfaces).

the other hand, face verification can be considered as one to one relation (1:1) between features and identity space, because in verification you receive just one answer to your question, a yes or a no. In other words, face recognition can answer the question "who are you?," rather than verification which can answer the question "Is this you?". Recent research, specifically research on unconstrained datasets, uses face verification for recognition purposes by applying the input probe image over the gallery dataset and checking which image would receive a yes answer at a certain rate of false alarm. A comparison between these systems is stated in [18]. The drawbacks of these systems can be listed as the low recognition rate for unconstrained single face image cases and the large execution time compared to the existing recognition systems.

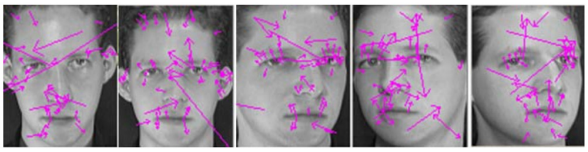
Additional systems utilized deep convolutional neural network (CNN) models to generate a verification system and employing it for face recognition. DeepFace [19], FaceNet [20], DeepID [21], DeepID2 [22], [23] and DeepID3 [24] are examples of these systems. Another approach uses higher dimension features of LBP or other techniques for verification systems as detailed in [25, 26].

There is also an approach that uses these high dimension spaces in the recognition process directly, as stated in [27] and shown in Figure 2.4 [27]. However, this paper mainly focuses on face alignment, and the technique developed in this paper did not recover the missing parts in the face after 3D transformation. Moreover, their work has been tested only on small datasets captured in controlled environments.



(a)

Figure 2.2: Example of LBP features extracted for samples from Extended Yale B+ for 3x3 window.



(a)

(b)

Figure 2.3: SURF features for (a) AT&T ORL dataset (b) Yale dataset.

2.2 Convolutional Neural Network (CNN)

Some machine learning techniques rely on designed heuristics for learning parameters. These heuristics require in depth consideration of the nature of the utilized system and data. Other techniques suggested the usage of self-learning features, a system that is closer to the human brain in nature. Convolutional Neural Network (CNN) is one of these self-feature extracting systems. CNN combines some architecture strategies close to human brain against shifting, scaling and distortion of inputs. These strategies are: local receptive fields, shared weights and spatial sup-sampling.

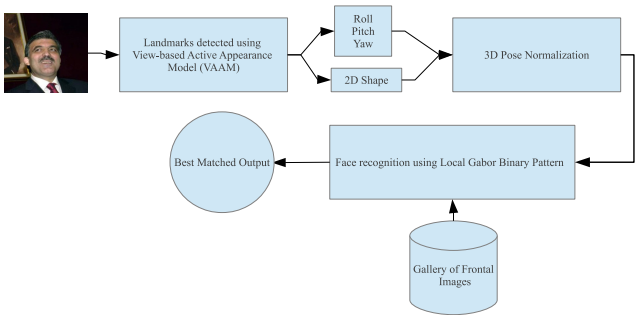


Figure 2.4: Automatic recognition system proposed by Asthana et al.

CNN is built basically on the concept of Neocognitron which has been introduced in [28]. This concept has been adjusted and used by [29] for characters recognition. The architecture used in this paper is shown in Figure 2.5 [29]. The basic concept of CNN is designing filters W_{ij} (filter number i) that can work as receptive fields in the level j and output certain features that will work as inputs to next level for other features extraction. The size of W_{ij} is called the kernel size of the weights. After a convolution process for the input with filters W_{ij}, sub-sampling process is applied to extract the important features from these outputs. This process can be applied several times until large number of meaningful features are extracted from the provided input (two times in Lenet5 [29]). Following this, layers of fully connected neural network can be used for dimension reduction of these large scale features to extract the most significant ones. Activation function can be used at any level of the network for rectification and adjustment. Gradient decent back-propagation algorithm has been used efficiently to learn these type of networks.

As of recent, the drawback of these type of networks involved the computational complexity, since these networks require millions of images for training, verification and testing to prevent over-fitting. In the last decade, significant improvements in the field of parallel implementation and general purpose graphical processing units GPGPUs justified the utilization of CNN, reducing the training time order to hours

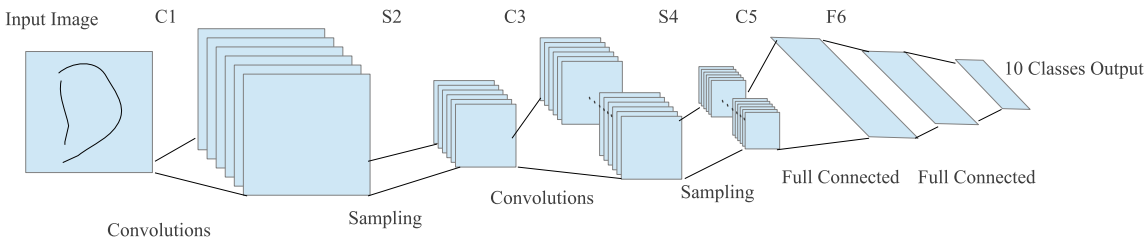


Figure 2.5: LeNet5 architecture.

and days as opposed to weeks and months. Therefore, as detailed in [30], a larger and deeper convolution network is designed that trained on ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) dataset (1.2 Million images), which has 1,000 classes to detect in the given input image. The architecture of this network is shown in Figure 2.6 [30]. In this design the authors used a different type of activation function called Rectified Linear Units (ReLUs) which is defined as $f(x) = \max(0, x)$. A normalization step and overlapping pooling are also used in this architecture.

Recently, with the advances in processing power, more deeper and more complex networks are designed and trained over larger datasets. The current state-of-the-art structure on CNN networks is detailed in [31]. In this network a deeper system with new inception modules is proposed to emulate the optimal local sparse structure of a convolutional vision network. Figure 2.7 [31] shows the structure of GoogleLeNet network designed in [31]. This network is the recipient of the first place award in the ILSVRC 2014 classification challenge with an error level of 6.67% for top 5 classification results.

2.3 Gradient Decent Back-Propagation

As stated in [32] and [33], assume that the neural system is structured as follows: The system has M layers feed-forward neural network with input p of size R, and

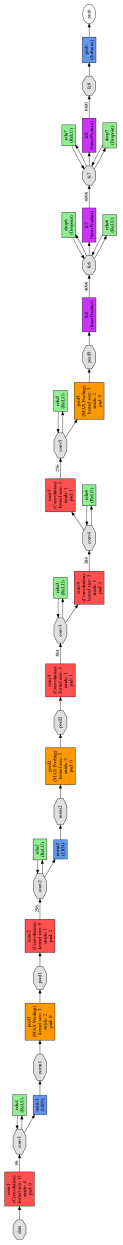


Figure 2.6: Implementation of Alexnet used with Imagenet dataset.

outputs y of size S_M with each layer having a number of nodes S_k, where, $k = 0, 1, 2, \ldots, M-1$ and the output of each layer is $a^k(j)$, where $a^0 = p$ and $j = 1, 2, \ldots, S_k$ with a final loss function $\mathcal{L}(W)$, where W represents the all network layers weights and biases vectors. Let $w^{k+1}(i,j)$ be the weight value between node j in layer k and node i in layer $k+1$ and $b^{k+1}(i)$ is the bias value of node i in layer

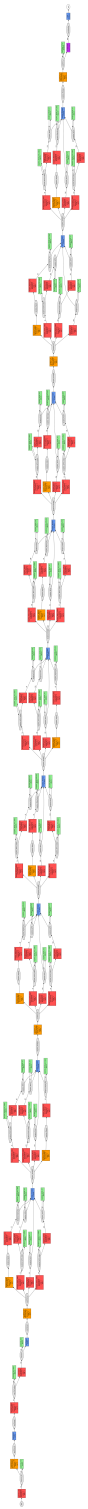


Figure 2.7: Implementation of GoogleLeNet architecture.

$k+1$. Assuming that the number of training pairs is Q (number of input-output pairs), the net input to node i in layer $k+1$ then becomes:

$$n^{k+1}(i) = \sum_{j=1}^{S_k} \left(w^{k+1}(i,j) . a^k(j) + b^{k+1}(i) \right), \tag{1}$$

and the output of this node will be

$$a^{k+1}(i) = f^{k+1}\left(n^{k+1}(i) \right), \tag{2}$$

where f^{k+1} is the activation function of this node.

Assuming that the final loss over all samples is:

$$\mathcal{L} = \sum_{q=1}^{Q} L_q. \tag{3}$$

Based on Gradient Decent Algorithm the weights and biases updates in each iteration are

$$\Delta w^{k+1}(i,j) = -\alpha . \frac{\partial L_q}{\partial w^{k+1}(i,j)}, \tag{4}$$

$$\Delta b^{k+1}(i) = -\alpha . \frac{\partial L_q}{\partial b^{k+1}(i)}. \tag{5}$$

where α is the learning rate. For simplicity a new term can be defined the sensitivity as

$$\delta^{k+1}(i) \equiv \frac{\partial L_q}{\partial n^{k+1}(i)}. \tag{6}$$

Based on this definition and equations (1) and (2), equations (4) and (5) can be reformulated as

$$\Delta w^{k+1}(i,j) = -\alpha . a^k(j) . \delta^{k+1}(i), \tag{7}$$

Algorithm 1 Gradient Decent Back-Propagation Algorithm

(1) Propagate the input forward using equations (1) and (2).

(2) Propagate the sensitivity back using equations (12), (8), (7) and (6).

(3) Update the weights and biases using equations (7) and (8).

(4) Repeat these steps until the stopping criteria are satisfied.

$$\Delta b^{k+1}(i) = -\alpha.\delta^{k+1}(i), \tag{8}$$

Also it can be shown that

$$\delta^k = \dot{F}^k\left(n^k\right).W^{k+1^T}.\delta^{k+1}, \tag{9}$$

where

$$\dot{F}^k\left(n^k\right) = \begin{bmatrix} \dot{f}^k\left(n^k(1)\right) & 0 & \cdots & 0 \\ 0 & \dot{f}^k\left(n^k(2)\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{f}^k\left(n^k(S_k)\right) \end{bmatrix}, \tag{10}$$

and

$$\dot{f}^k\left(n^k(i)\right) = \frac{df^k\left(n^k(i)\right)}{dn^k(i)}. \tag{11}$$

This recurrent relation will work until the final layer where the initial value of δ^Mequal

$$\delta^M = \dot{F}^M\left(n^M\right).\frac{\partial L_q}{\partial n^M}, \tag{12}$$

The steps of the algorithm can be listed as in Algorithm 1.

CHAPTER 3: FACE RECOGNITION SYSTEM FOR LARGE SCALE DATASETS AND UNCONSTRAINED FACES

3.1 Proposed System

The first part of this dissertation presents a face recognition system for unconstrained captured faces and large scale datasets based on Hierarchical Sub- Graph Selection (HSGS) Algorithm for faces grouping. The proposed system operates as in the following. The first step is the face detection which requires checking the input image for faces and determining the resolution of the detected faces, if any. If the resolution of these faces is low or the faces could not be detected, a common occurrence in surveillance camera images, the input image is then applied to super- resolution module to increase face resolution. Next, face landmarks are detected to aid in face pose estimation. Following this, in order to improve the recognition process, the 3D aligning algorithm is applied to the face area to produce a canonical face position. The image is then fed to the HSGS algorithm to detect the top 50 matches which provides high recognition rate for rank 50. To extract better rank 1 from these top 50 faces, a face verification system is applied consecutively to obtain the best match. The block diagram for the proposed system is provided in Figure 3.1. A detailed explanation of each block is included in the following sections.

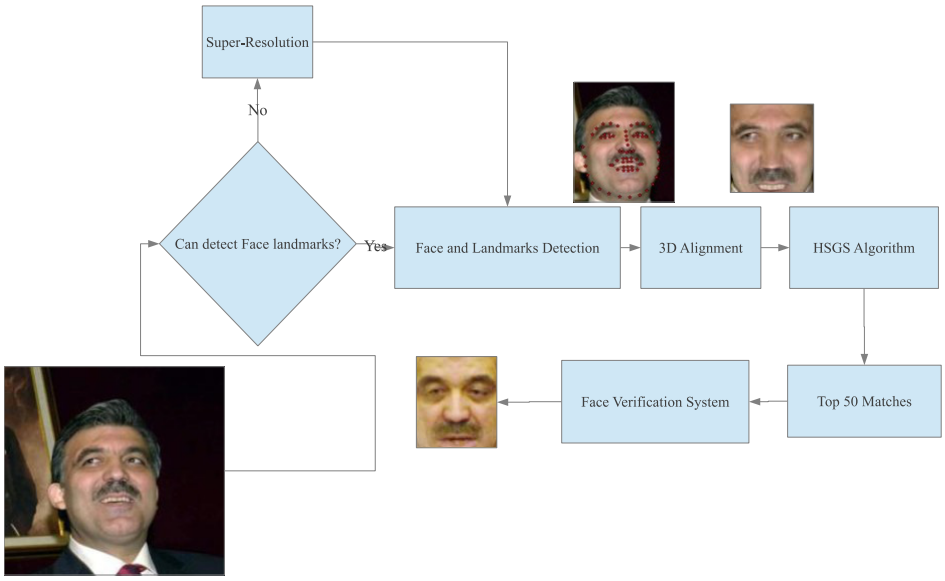


Figure 3.1: Proposed face recognition framework.

3.2 Super-Resolution Image

As mentioned previously, Super-Resolution algorithm is used for enhancing the image resolution, providing additional details of the input image. In this section, a super-resolution image algorithm based on Convolutional Neural Network (CNN) is employed as described in [34]. The system first generates a low resolution higher dimension image from the input image using bicubic interpolation (Y). This image is then applied to CNN network structure as shown in Figure 3.2 to increase the image accuracy and to generate a higher resolution image $\left(\bar{Y}\right)$ that should be close to the original image (X). The following subsections details the steps of this process. As stated in [34], this algorithm outperforms all other state-of-the-art algorithms used for patch-based single image super-resolution. This superior performance is achieved since the CNN network is able to learn and improve the performance with more iterative training rather than the other algorithms that use closed formulas for patch extraction prohibiting improvements via additional samples or trainings.

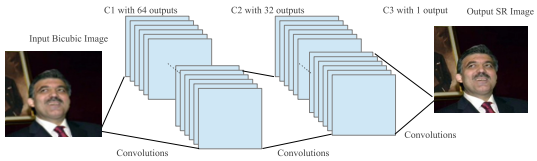


Figure 3.2: Super-Resolution using CNN (SRCNN).

3.2.1 Patch extraction and representation

In this step, the main task is to extract n_1 filter that will work as patch extraction from the input image Y. The size of each filter is $f_1 \times f_1$. This process can be described by equation (1)

$$F_{1i}(Y) = \max(0, W_{1i} * Y + B_{1i}), \tag{1}$$

where W_{1i} and B_{1i} represents the filters and biases respectively and $i = 1, 2, 3, \ldots, n_1$. Rectified Linear Unit (ReLU) activation function is used over this step output.

3.2.2 Non-linear mapping

This step can be applied once or multiple times to enhance and to improve the image patches extracted from the previous level. The filters for this step will be n_2 filter of size $f_2 \times f_2$ each (the initial value used for f_2 is 1), and will be applied to the n_1 patches extracted from the last level. Equation (2) describes this process.

$$F_{2ij}(Y) = \max(0, W_{2j} * F_{1i}(Y) + B_{2j}), \tag{2}$$

where W_{2j} is the $f_2 \times f_2$ filter, B_{2j} is the bias and $j = 1, 2, 3, \ldots, n_2$.

3.2.3 Reconstruction

This final step is responsible for reconstructing the high-quality image from the nonlinear extracted patches. This process averages these patches to form the final super-resolution image. The average process is also constructed as a convolutional layer with an average like filter. The size of these filters will be $f_3 \times f_3$ and they will work over the n_2 output of the previous step resulting in the identical number of channels as the original input image. The final output of the network will be the high resolution image $\bar{Y}$ as shown in equation (3).

$$\bar{Y} = G(Y) = W_3 * F_{2ij}(Y) + B_3, \tag{3}$$

where W_3 is the $f_3 \times f_3$ filter and B_3 is the bias vector and has the same size as the number of image channels.

3.2.4 Loss function

To learn this network, Stochastic Gradient Descent Back-propagation algorithm will be used to find W filters and B vectors. Mean Squared Error (MSE) between the reconstructed super-resolution images, $\left(\bar{Y}_i\right)$, and the original images, (X_i), will be used as the loss function as described in equation (4). This function will be minimized to obtain the optimal value of P that minimizes its value.

$$\mathcal{L}(P) = \frac{1}{n} \sum_{i=1}^{n} \|G(Y_i; P) - X_i\|^2, \tag{4}$$

where $P = \{W_1, W_2, W_3, B_1, B_2, B_3\}$

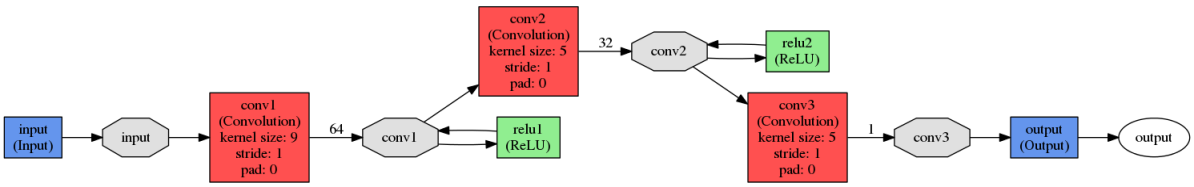


Figure 3.3: Implementation of the SRCNN.

3.2.5 Super-Resolution Implementation and Results

The super-resolution module has been implemented using Python wrapper of Caffe and Theano libraries as shown in Figure 3.3 with the following parameters: $f_1 = 9$, $f_2 = 5$, $f_3 = 5$, $n_1 = 64$, $n_2 = 32$ with upscaling factor of 3. Initially, the network is applied to Y component only in the YC_bC_r image presentation. Further improvement is achieved by applying the network on the 3 channels of the presentation for different presentations (YC_bC_r and RGB). A detailed explanation of this improvement is presented in Chapter 4. It should be noted that the images obtained in Figure 3.4 are the obtained samples resulting from the application of the module. The average Peak Signal to Noise Ratio (PSNR) for this technique is $PSNR = 33\,dB$ rather than $PSNR = 30\,dB$ in Bicubic, which means that the average improvement in the $PSNR = 3\,dB$ over the regular Bicubic scaling.

3.3 3D Face Alignment

Most images captured in unconstrained environments (in the wild), especially via surveillance cameras, do not provide face images in canonical frontal pose. To transform the captured image to its canonical pose a 3D aligning system is required. The aligning system used in this work is based on face frontalization algorithm presented in [35]. The system is based on a single reference 3D model in the canonical form, and the target of the system is to project back the input query image to this reference

Figure 3.4: Samples of images used in the experiment. From left to right: The original small image, The original large size, Bicubic generated image, SRCNN generated image.

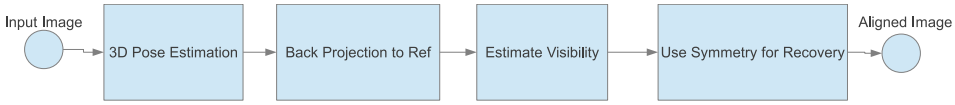


Figure 3.5: 3D face alignment system.

pose. Block diagram of the system is provided in Figure 3.5.

Figure 3.6 [35] shows an example of the alignment process. In steps (a) and (b) the face is detected and the landmarks are located with any state-of-the-art algorithm for landmarks detection. Then the same algorithm used with the query image should be used with the reference model to find its landmarks as shown in step (c). Following this in (d) a camera calibration process is applied to estimate the extrinsic and intrinsic camera parameters for 3D pose estimation of the query image. Back projection step is shown in step (e) to place the image in the canonical pose. Because of lack of view on some spots due to the original pose, an estimate visibility will be applied to projected frontal view to estimate the places of poor frontal projection as shown in step (f). Benefiting from the symmetric nature of faces, a final step is applied to

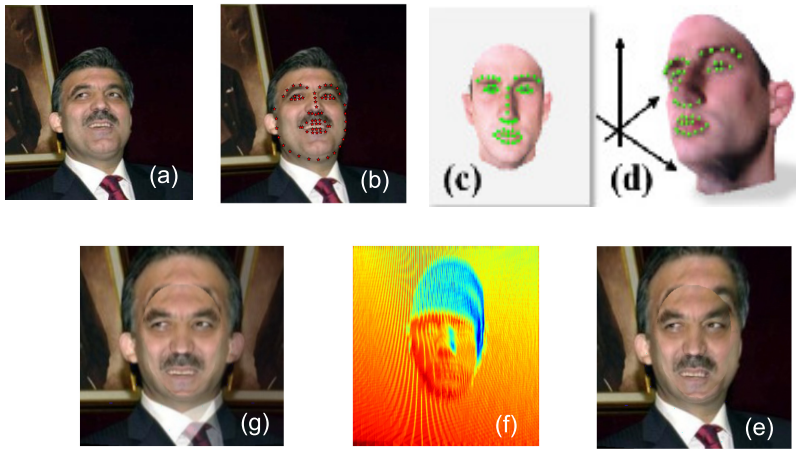


Figure 3.6: Example steps in the 3D face alignment process.

estimate these poor areas as shown in step (g).

3.3.1 3D Face Alignment Implementation

This algorithm has been implemented using Python wrapper of OpenCV and dlib libraries, and this implementation has been applied, especially on the unconstrained captured faces datasets, to enhance face alignment in the recognition framework. The recognition results of aligned images on unconstrained faces are shown in the following chapter. It is important to note that the clear details of the output image are factors of the initial angle. That is, the algorithm may fail to recover missing parts or show face details if the face yaw and pitch angles rotations exceeds a 60 degree (60^o). The algorithm may also fail if the input face resolution is much higher than the reference model. Therefore, additional adjustments are required to increase the robustness of the algorithm.

3.4 Hierarchical Recognition Technique

As explained in Chapter 2, most unsupervised recognition algorithms suffer from rate degradation when the database size is large, especially for unconstrained faces. Therefore, some researches proposed the utilization of dataset partitioning with hierarchical recognition. However, none of these researches used unsupervised grouping. This dissertation proposes the utilization of unsupervised metrics for a Hierarchical Sub-Graph Selection algorithm for image datasets partitioning and unsupervised recognition. The details of this algorithm with corresponding results are presented in Chapter 5.

3.5 Face Verification Module

Using similarity metric is proven to be an efficient method for face verification especially if the number of images per class is limited. In this work, similarity metric method is used with Convolutional Neural Network (CNN) to perform a Siamese Network that will be applied in the final step of the proposed system to increase rank 1 recognition rate and to improve the recognition time.

Using Siamese Network for face verification has been introduced in [36], where two input images X_1 and X_2 are applied to the same nonlinear mapping G_W to extract the main features that minimize the main energy function E when X_1 and X_2 belong to the same person and maximize it when X_1 and X_2 belong to different persons. The typical structure for this network is shown in Figure 6.6 as explained in [36]. The formal definition of the function E can be expressed as in equation (20)

$$E\left(W, X_1, X_2\right) = \left\| G_W\left(X_1\right) - G_W\left(X_2\right) \right\|, \tag{5}$$

where W are the shared weight filters between the two input images.

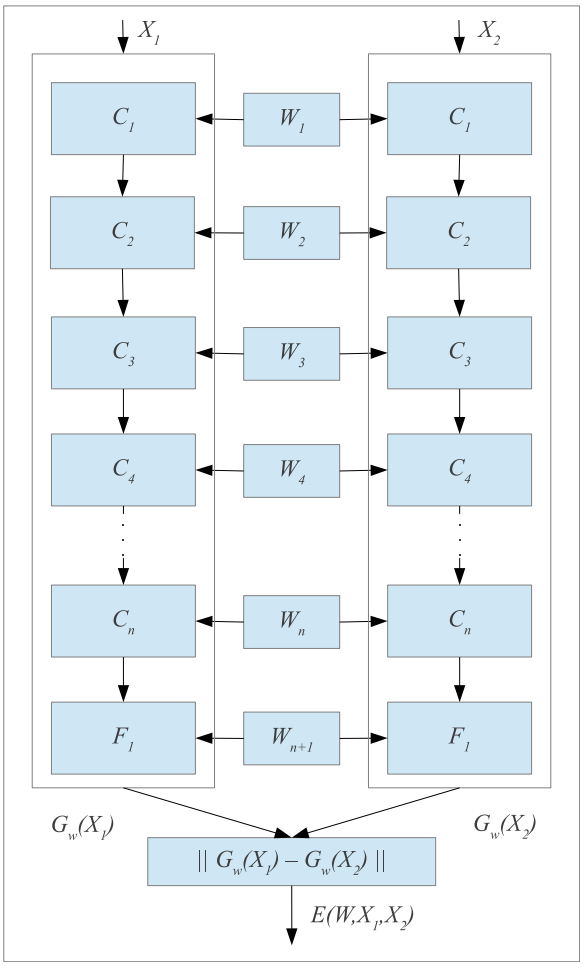


Figure 3.7: Typical structure of siamese network.

To achieve this goal for the E function, the loss function should monotonically increase with same person pairs energy and monotonically decrease with different persons pairs energy. Based on this logic, the final loss function will be formed as in equations (21), (22), (23), (24), and (25)

$$\mathcal{L}(W) = \sum_{i=1}^{N} L\left(W, (Y, X_1, X_2)^i\right), \tag{6}$$

$$L\left(W, (Y, X_1, X_2)^i\right) = Y \cdot L^s\left(E(W, X_1, X_2)^i\right) + (1 - Y) \cdot L^d\left(E(W, X_1, X_2)^i\right), \tag{7}$$

$$L^s\left(E\left(W,X_1,X_2\right)^i\right)=\frac{2}{Q}\left(E\left(W,X_1,X_2\right)^i\right)^2, \tag{8}$$

$$L^d\left(E\left(W,X_1,X_2\right)^i\right)=2Q\cdot e^{\left(-\frac{2.77}{Q}E(W,X_1,X_2)^i\right)}, \tag{9}$$

$$Y=\begin{cases}1 & X_1\equiv X_2\\ 0 & X_1\not\equiv X_2\end{cases}. \tag{10}$$

where N is the number of training samples, Y is equal to 1 if X_1 and X_2 for the same person and 0 if X_1 and X_2 are for different persons, L^s is the loss function in the case of same person, L^d is the loss function in the case of different persons and Q is a constant representing the upper bound of E.

Because the energy is monotonically changing for both L^s and L^d, the optimization of the loss function can be easily achieved using simple gradient decent algorithm, and the weights W can be learned using back-propagation algorithm.

3.5.1 Siamese Network Implementation and Results

The design of the proposed Siamese structure is implemented using Caffe library as shown in Figure 3.8. The network consists of two identical Convolutional Neural Networks (CNN) both shared the same parameters followed by full connected feed-forward neural networks (FCFNN). The parameters of the network are provided in the following:

- Layer 1 CNN: Kernel size 3, Output layers 32,
- Layer 2 PReLU activation function,
- Layer 3 CNN: Kernel size 3, Output layers 64,
- Layer 4 PReLU activation function,

- Layer 5 Max Pooling Layer of kernel size 2,
- Layer 6 CNN: Kernel size 3, Output layers 32,
- Layer 7 PReLU activation function,
- Layer 8 CNN: Kernel size 3, Output layers 64,
- Layer 9 PReLU activation function,
- Layer 10 Max Pooling Layer of kernel size 2,
- Layer 11 CNN: Kernel size 3, Output layers 32,
- Layer 12 PReLU activation function,
- Layer 13 CNN: Kernel size 3, Output layers 64,
- Layer 14 PReLU activation function,
- Layer 15 Max Pooling Layer of kernel size 2,
- Layer 16 CNN: Kernel size 3, Output layers 32,
- Layer 17 PReLU activation function,
- Layer 18 CNN: Kernel size 3, Output layers 64,
- Layer 19 PReLU activation function,
- Layer 20 Max Pooling Layer of kernel size 2,
- Layer 21 CNN: Kernel size 3, Output layers 32,
- Layer 22 PReLU activation function,
- Layer 23 CNN: Kernel size 3, Output layers 64,

- Layer 24 PReLU activation function,
- Layer 25 Max Pooling Layer of kernel size 2,
- Layer 26 Dropout 50%,
- Layer 27 FCFNN: Output nodes 4000, PReLU activation function,

The 4,000 output features from each branch are then applied to the loss function as described in the previous subsection. Stochastic Gradient Decent algorithm is used with patches for parallel execution to learn network parameters. Verification rate of this network over 3D aligned LFW dataset [37][38] using unrestricted protocol without any outside data is approximately 99.9% which comparable to the state-of-the-art results published on this dataset[1] . Figure 3.9 shows the receiver operating characteristic (ROC) and Precision-Recall curves for the implemented network.

[1] For latest results on LFW dataset, visit http://vis-www.cs.umass.edu/lfw/results.html

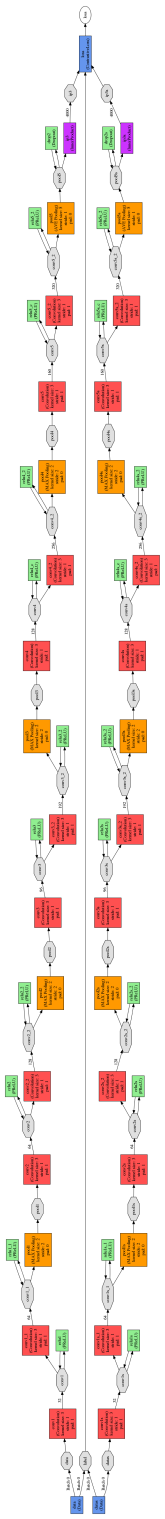
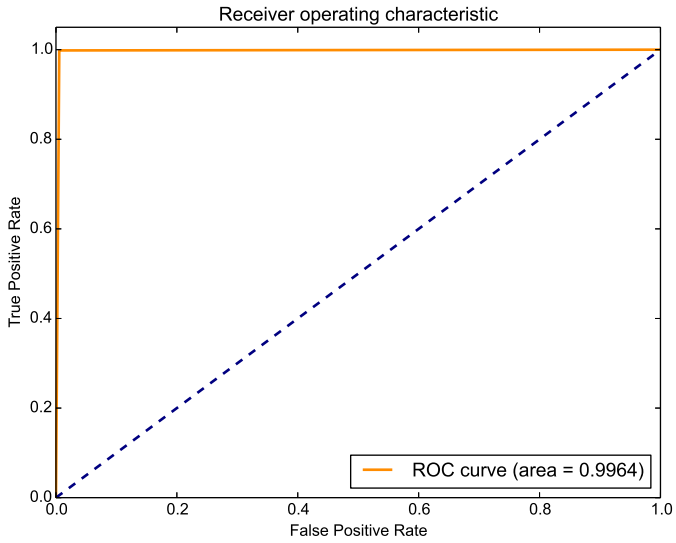
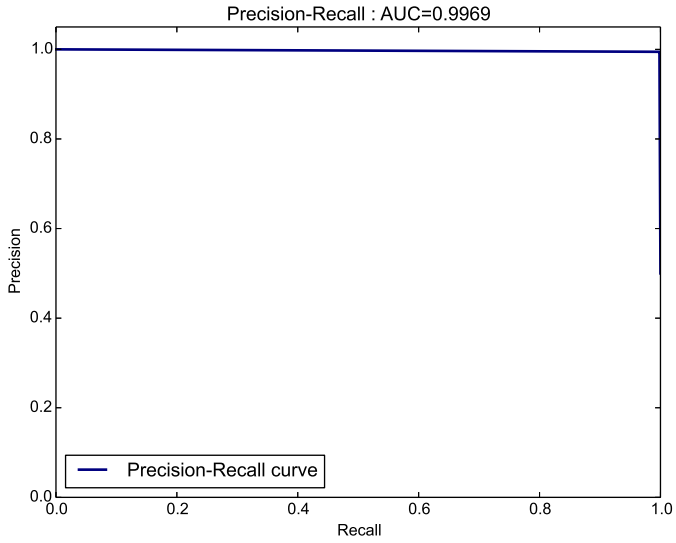


Figure 3.8: Implemented Siamese network.



(a)



(b)

Figure 3.9: Implemented Siamese Network (a) ROC curve, (b) Precision-Recall Curve.

CHAPTER 4: UNSUPERVISED FACE RECOGNITION IN THE WILD USING HIGH DIMENSIONAL FEATURES UNDER SUPER RESOLUTION AND 3D ALIGNMENT EFFECT

4.1 Introduction

Majority of the surveillance cameras are installed outdoors. Unlike their counterparts, the indoor images, the images captured via outdoor cameras are impacted by external factors caused by the surrounding environment. Often referred as "images in the wild", these images require unique procedures when they are used in facial recognition since their size and resolution has a direct impact on the recognition accuracy. Previous facial recognition literature offers a number of studies concentrating on multi-frame based super resolution construction of the low resolution face images [39, 40, 41, 42]. Majority of these however, deal with testing the performance of traditional face recognition techniques on lower and super resolution faces which are derived from multi-frame videos. This approach becomes prohibitive in practice when a multi-frame video is not available and the face recognition problem needs to be solved based on a single query image. There are additional similar studies which utilize a single image based super-resolution algorithm to study the performance of face recognition algorithms on different face resolutions [43, 44]. However, these stud-

ies utilize test datasets that contain images captured in controlled environments and also fall short in analyzing the performance of face recognition using high-dimensional features.

Aiming at filling these gaps, this study investigates the performance of unsupervised face recognition on the labeled faces in the wild (lfw) dataset [37, 38] using a single image super-resolution (SR) algorithm. To achieve this, the effect of the algorithm on high dimensions extracted features used in the face recognition process is measured. All the faces included in the dataset are 3D aligned and frontalized using face frontalization algorithm proposed in [35].

The main contribution of this research can be summarized as in the following.

- Applying high dimensional features (Local Binary Pattern (LBP) and Speed Up Robust Features (SURF)) and Multi-Scale version from these features on captured faces in the wild.
- Using these calculated features in unsupervised closed set face recognition protocol.
- Comparing the effects of single image super-resolution algorithm and bicubic scaling on unsupervised face recognition in the wild.
- Examining the effect of the order of applying face frontalization and image sharpness (super-resolution) processes to determine the order with the best recognition rate.

Following sections provide detailed explanation regarding the super-resolution algorithm utilized in this research. A discussion regarding high-dimensional features along with the best performing one for unsupervised learning is provided. This is followed by the description of the proposed experiments and the techniques used in it are

presented, followed by a results section that explains the collected results. Finally, conclusions and discussion are presented.

4.2 Single Image Super-Resolution

Super-Resolution algorithm is used for enhancing the image resolution to obtain additional details from a given image. This works utilizes a Convolutional Neural Network (CNN) based super-resolution image algorithm as described in [34]. The system first generates low resolution higher dimension image from the input image using bicubic interpolation. This image is then applied to a CNN network structure as shown in Figure 4.2 to improve image peak signal to noise ratio (PSNR), and to generate a higher resolution image similar in quality to the original image. Implementing this system as CNN makes it superior to other SR techniques that generates mapping from low to high resolution images due the simplicity of the system and its relatively higher PSNR value.

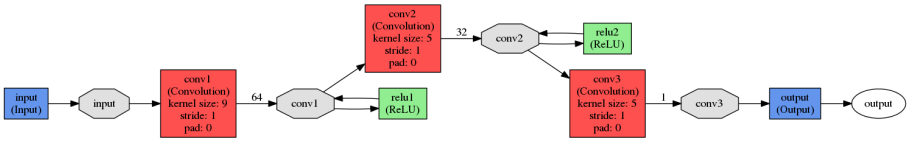


Figure 4.1: Super-Resolution using Convolutional Neural Network (SRCNN) algorithm used in the test.

4.3 High Dimensional Features

Unsupervised face recognition has recently gained increasing popularity due its ability to handle unlabeled faces. Unsupervised face recognition is also able to increase the dataset without retraining the classifier, especially in closed datasets as in [45, 10, 46]. Previous research on high dimensional features has provided remarkable

results in face recognition and verification especially with supervised learning as in [47, 26]. However, these features have yet to be sufficiently explored using unsupervised techniques. This section demonstrates the utilization of two of these features with an unsupervised metrics for closed set protocol on the lfw dataset.

4.3.1 Local Binary Pattern (LBP) Features

In [10], LBP features has shown remarkable unsupervised face recognition results for faces in controlled environments. The LBP features of an image is calculated as:

$$LBP_{B,R} = \sum_{b=0}^{B-1} \delta\left(I_b - I_c\right) 2^b \tag{1}$$

where

$$\delta\left(x\right) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

which can represented by 8 bit representation with 256 maximum possibilities. This number would reduce to 59 when only uniform patterns are considered. Therefore, the image can be represented by 59-dimensional histograms using the following equation:

$$H(i) = \sum_{x,y} \alpha\left(LBP^{u2}_{B,R}(x,y)\right) \tag{2}$$

where

$$\alpha\left(x\right) = \begin{cases} 1 & x = i \\ 0 & otherwise \end{cases}, \; i = 0, 1, 2, \ldots, 58$$

When the images are divided into blocks, the equation for the histogram will become:

$$H_j(i) = \sum_{x,y} \alpha\left(f_j\left(x,y\right)\right) \tag{3}$$

where $f_j\left(x,y\right)$is the $LBP^{u2}_{B,R}$ features in block j of the image and $j = 0, 1, \ldots, M-1$. Here M indicates the number of blocks in the image. The histogram then will be a concatenation of all block histograms (4)

$$H_{total} = |H_0\left(i\right)\ H_1\left(i\right)\ H_2\left(i\right)\ \ldots H_M\left(i\right)| \tag{4}$$

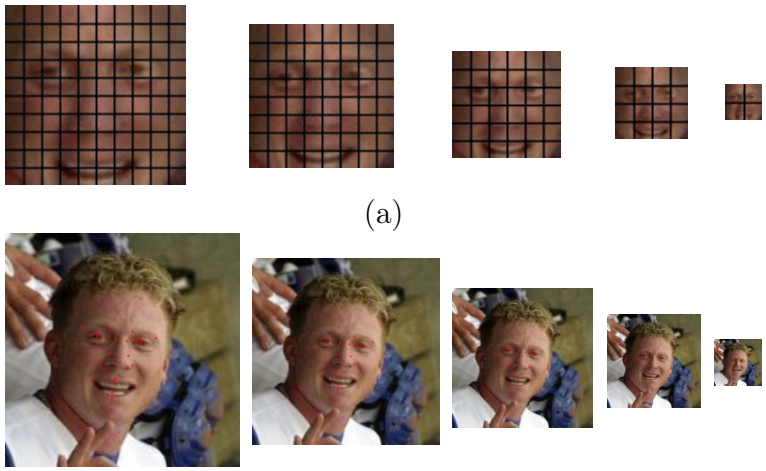
In this test, the Chi square metric is used with the extracted features from the lfw dataset as in equation (5).

$$\chi^2\left(X,Y\right) = \sum_{i,,j} \frac{\left(x_{i,j} - y_{i,j}\right)^2}{x_{i,j} + y_{i,j}}, \tag{5}$$

where,X and Y are the histograms to be compared, i and j are the indices of the i-th bin in histogram corresponding to the j-th local region.

In this test, three types of LBP features are demonstrated. First, regular uniform LBP features are extracted from frontalized faces by dividing the 90x90 face into 10x10 blocks, each 9x9 pixels, and calculating (8,2) ($LBP^{u2}_{8,2}$) neighborhoods for each block as in [10]. Following this, the histograms of all blocks are concatenated together to form a single vector representation for the face image, which can be used in equation (5). The output vector of this calculation will be of length 5900.

The second type of LBP is a Multi-Scale representation. Here, the frontalized face is scaled down 5 times, and for each scale the image is divided to 10x10 blocks of 9x9 pixels each as shown in Figure 4.2 a, and again $LBP^{u2}_{8,2}$ histogram is again calculated for each block at each scale as shown in Figure 4.3 and all histograms are concatenated together to form a vector representation for the face, which will be in a length of 12980.



(a)

(b)

Figure 4.2: Two LBP features a)Multi-Scale LBP b)HighDimLBP.

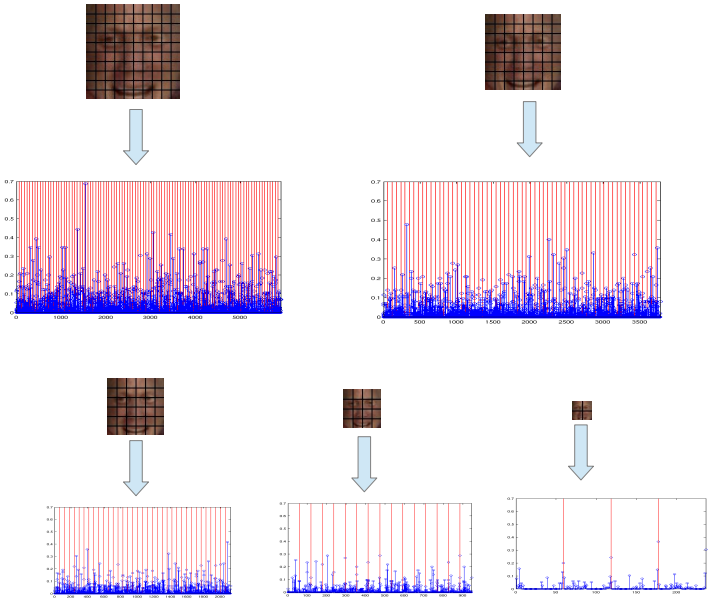


Figure 4.3: Multi-Scale LBP and the histograms calculated from each block at each scale.

The final LBP type is the HighDimLBP introduced in [26],where an accurate landmark detection technique is used to detect landmarks of faces as opposed to frontalizing face images. Each landmark in the 300x300 image a grid of 40x40 centered at each landmark point is then constructed and $LBP^{u2}_{8,2}$ is calculated over each 10x10 pixels

block as shown in Figure 4.2 b, All histograms from all blocks for all landmark points on the 5 different scales are concatenated together to form a vector representation of the face image. The length of this vector for a single image is 127440, which is computationally prohibitive. Similar to [47, 26], to decrease the computational complexity, the length of the vector is reduced to 400 via principle component analysis (PCA) when needed.

A detailed analysis is conducted among these three types to obtain the most effective technique. The results of the analysis are provided in section 5.

4.3.2 Speed Up Robust Features (SURF)

Scale Invariant Feature Transform (SIFT) features have previously been utilized for face recognition using support vectors machines with supervised learning as in [12], or with unsupervised metric learning as in [13, 46]. However, SURF features have only been used for unsupervised face recognition and produced comparable results as reported in [46]. To analyze the method's effectives on the facial images captured in the wild, in this study, SURF features are applied to super-resolution and frontalization processed faces. A new multi-scale version of this technique is also introduced to improve the recognition performance.

The key-points for the SURF algorithm are predefined in the odd number of pixels for each row and column with a radius of 2. Since the face is frontalized, only upright SURF (U-SURF) is used with its 64 dimensional feature vector per key point. In the matching step, the face image is divided into 5x5 pixel blocks.

The matching metric for the calculated features is based on equation (6). Figure 4.4 shows an example of a single scale matched and unmatched pairs.

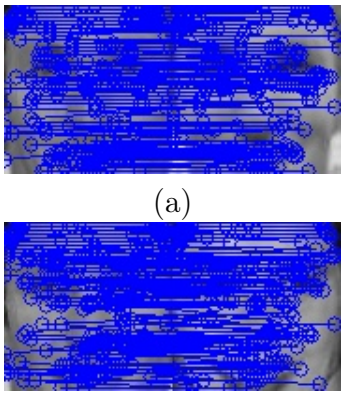
$$m(Z) = arg\max_{c}\left\{\max_{n}\left\{g(Z, L_{n,c})\right\}\right\} = arg\max_{c}\left\{\max_{n}\left\{\sum_{z_i \in Z}\beta(z_i, L_{n,c})\right\}\right\} \quad (6)$$

where

$$\beta\left(z_i, L\right) = \begin{cases} 1 & \min_{l_j \in L}\left\{d\left(z_i, l_j\right)\right\} < \varepsilon . \min_{\acute{l}_j \in L}\left\{d\left(z_i, \acute{l}_j\right)\right\} \\ 0 & otherwise \end{cases}$$

where

ε is the nearest neighbor ratio scaling parameter and will set to $\varepsilon = 0.5$ and $\acute{l}_j$ is the neighbor point to l_j.



(a)

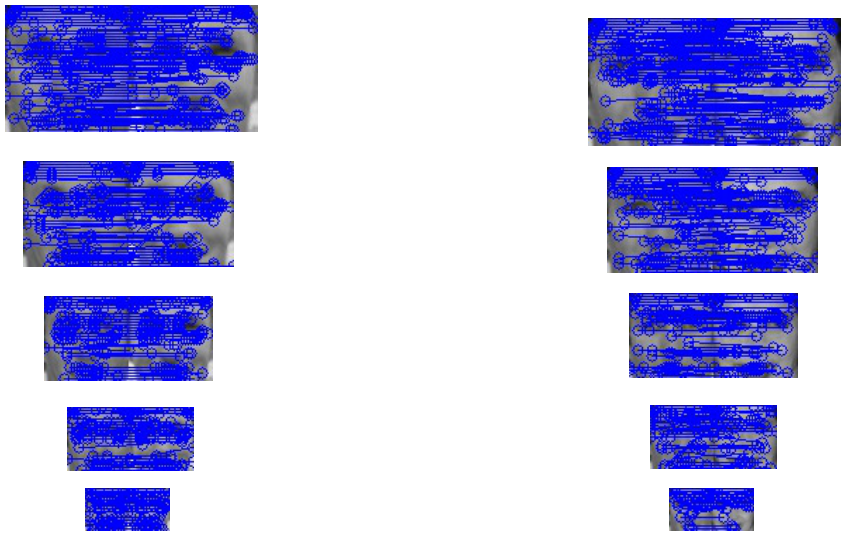
(b)

Figure 4.4: U-SURF features matching at one scaling level for a) Matched Pair b) Unmatched Pair.

Since the key points are predefined (i.e., SURF steps do not include multi-scale calculations), a multi-scale version of the features is introduced and tested for its performance versus the one scale version as shown in Figure 4.5. In this case the matching metric will be:

$$m\left(Z\right) = arg\max_{c}\left\{\max_{n}\left\{\sum_{s}\sum_{z_i \in Z}\beta\left(z_i, L_{n,c}\right)\right\}\right\} \tag{7}$$

where s is the number evaluation scales.



(a) (b)

Figure 4.5: Multi-Scale U-SURF features matching for a) Matched Pair b) Unmatched Pair.

4.4 Experiment Description

This work proposes three experiments and examines their impact on image super-resolution and the order of applying it with frontalization in unsupervised face recognition process based on the features described in section 3. These experiments are detailed in the following:

(1) Apply face frontalization first. The work flow of this experiment is shown in Figure 4.6 a, and can described in the following steps:

   (a) Detect and frontalize face from the original sized image (250x250 in this case).

   (b) 2. Scale down the face image by scale of 3 (if the frontalized face image is 90x90, it will be reduced to 30x30, an acceptable size for face detection techniques, assuming face detection with an identical size[1]).

[1] Minimum detection size of Haar Cascade classifier for face detection is 24x24

(c) Scale up the face image again by scale of 3 using bicubic technique.

(d) Apply the SRCNN algorithm to the scaled face to generate a super-resolution version.

(e) Extract features from the SR-image:

  (i) For uniform local binary pattern (LBP), by dividing it into 10x10 blocks and concatenating the histograms of all blocks together. This step will be applied on both bicubic and super-resolution scaled faces to compare the performance of the recognition process.

  (ii) For speed up robust features (SURF), the upright SURF (U-SURF) should be calculated using the pre-defined key-points as described in section 3. The image then will be divided into 18x18 blocks and matching will be applied between every two similar place blocks.

(f) For Multi-Scale Features:

  (i) For LBP, the face image will be scaled down by five scales as shown in Figure 4.3. The histograms of all blocks and scales are then concatenated together. This step will then be repeated for both bicubic and super-resolution scaled faces to compare the performance of the recognition process.

  (ii) For SURF, the face image will be scaled down by five scales as shown in Figure 4.5. The features will then be calculated at all key points in all of the five scales. Equation (7) will be applied to calculat matching sum in all scales.

(g) Calculate metric distance for each features:

  (i) χ^2distances as shown in equation (5) between extracted features to find the minimum distances between the query images and the prob

ones.

(ii) Nearest neighborhood algorithm will be used to find the matching of the key-points within the similar block location in both the query image and the dataset. Then equation (6) will be applied to obtain the maximum matched pair.

(2) Process face images first before frontalization. The work flow of this experiment is shown in 4.6 b, and can described as in the following steps::

(a) Scale down the face image by scale of 3.

(b) Scale up the face image again by scale of 3 using bicubic technique.

(c) Apply SRCNN algorithm to the scaled image to generate a super-resolution version.

(d) Extract frontalized faces from both bicubic images and super-resolution ones to compare performance.

(e) Calculate features and distances as described in steps e to g in experiment 1.

(3) No scaling up and down of the given images, but will apply the frontalization algorithm followed by SRCNN as shown in Figure 4.6. The steps of this approach is described in the following:

(a) Detect and frontalize face from the original sized image (250x250 in this case).

(b) Apply SRCNN algorithm to the frontalized 90x90 face to generate a super-resolution version.

(c) Calculate features and distances as described in steps e to g in experiment 1.

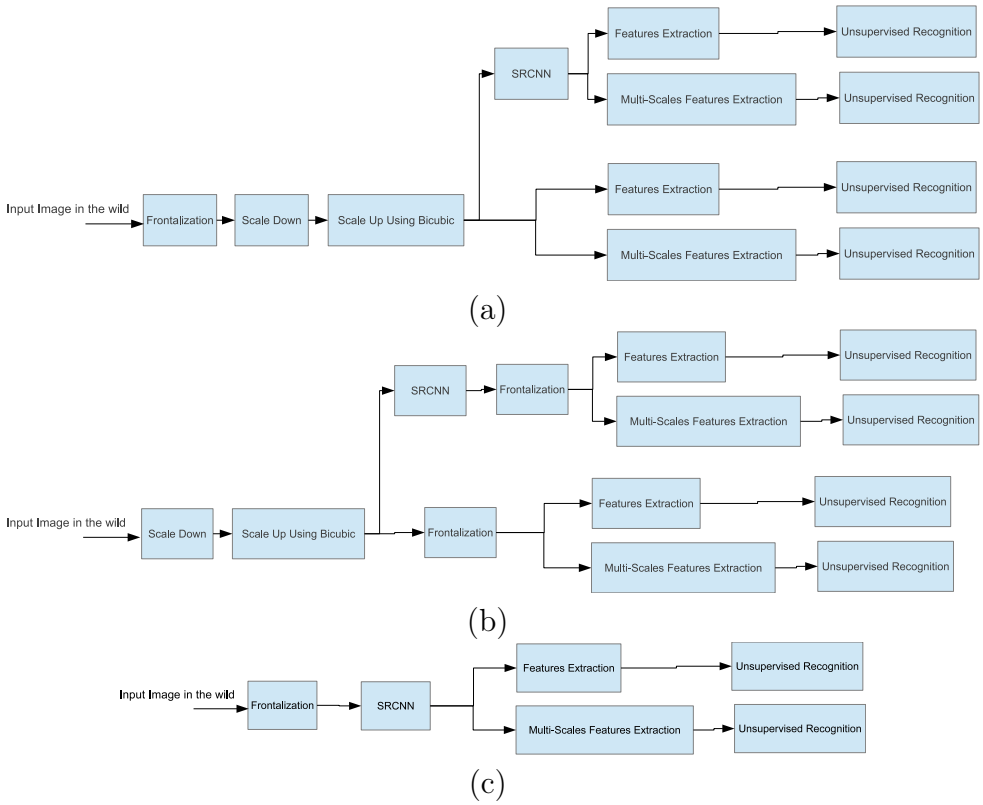


Figure 4.6: Proposed experiments a)Frontalization first b)Scaling and SR first c)Frontlization with SR without scaling.

	Rank 1 rate (%)
LBP	25.09
Multi-Scale LBP	26.51
HighDimLBP	26.30
HighDimLBP+PCA [47]	16.50

Table 4.1: Average rank 1 recognition rate using different LBP features.

4.5 Results

The proposed comparison and experiments have been tested on label faces in the wild dataset (lfw) [37, 38] using closed set face recognition protocol proposed in [47]. In this protocol, 10 groups are extracted from the entire dataset. Here, each group has two sets, viz., gallery set and genuine prob set. Both the gallery and prob belong

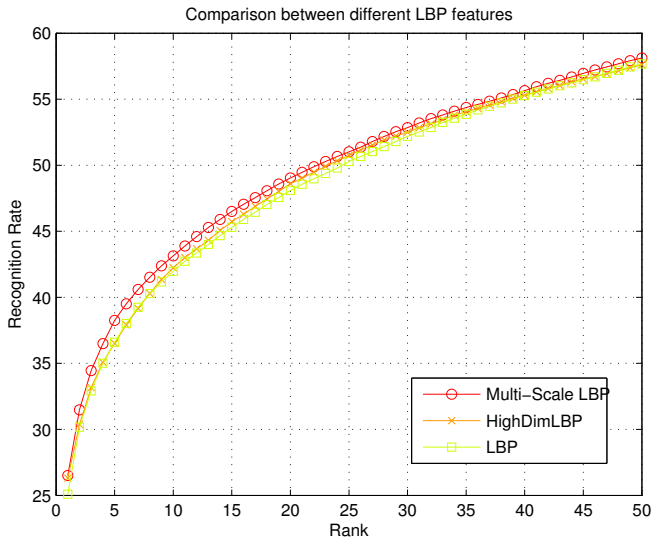


Figure 4.7: Average percentage recognition rate for 3 different LBP features.

	Rank 1 rate (%)
SURF	16.10
Multi-Scale SURF	24.60

Table 4.2: Average rank 1 recognition rate using different SURF features.

to the same 1000 different persons. Each gallery set contains 1000 images (one image per person in the gallery set) and the size of the prob set vary from one group to another with an average of 4500 images for the same 1000 persons in the gallery set. The calculated recognition rates in this chapter are the average recognition rates over the all 10 protocol groups.

In this work, the faces are detected using Histograms of oriented gradients (HoG) algorithm proposed in [48] and implemented in python. Then for each detected face, a landmark detection algorithm based on regression tree is used for face landmark detection as in [49] and implemented in python[2] . In some cases in experiment 2, due to the effect of image scaling, HoG based face detection algorithm failed to detect

[2] Python wrapper for dlib and OpenCV libraries

faces. Therefore, an additional backup face detection algorithm based on Adaboost Haar Cascade [50, 51] is used when no face is detected in the image [3] .

First, a comparison among the 3 different types of LBP features has been applied to this dataset on the frontalized faces without the effect of super-resolution algorithm and Chi square metric has been used as an unsupervised face recognition metric. As shown in Figure 4.7, the Multi-Scale LBP features outperforms all other LBP types, especially the method of using HighDimLBP+PCA as presented in [47]. As also shown in table 4.1, both Multi-Scale LBP and HighDimLBP with Chi square distance have comparable recognition rates. It is also important to note that the computation time of Chi square distance for HighDimLBP is significantly high compared to other LBP types due to the length of the vector representation.

Another comparison among SURF features (single vs multi scale) has been evaluated to investigate the benefits of multi-scale features. For this test, a nearest neighborhood based on k-dtree has been used with a tree depth of 5. After obtaining matching points, transformation matrix has been evaluated using RANSAC algorithm to find the homography between matching corresponding pairs key-points. The matching metric is the total number of matching points which satisfy this transformation. As shown in table 4.7, Multi-Scale SURF shows better performance than single scale SURF with results comparable with LBP features.

For the three experiments, the super-resolution based on convolutional neural network (SRCNN) algorithm is implemented using Caffe library and tested using MATLAB. As mentioned in [34], SRCNN is only applied on the y component of the ycbcr domain since it is the one with the high frequencies. In this test, the SR algorithm is applied on the y component of the ycbcr domain in addition to the three channels of the RGB domain to compare the performance of the algorithm when used with the

[3] Adaboost Haar Cascade classifier is known to have higher face detection rate but with large number of false positives [50, 51]

color components rather than only on the y component.

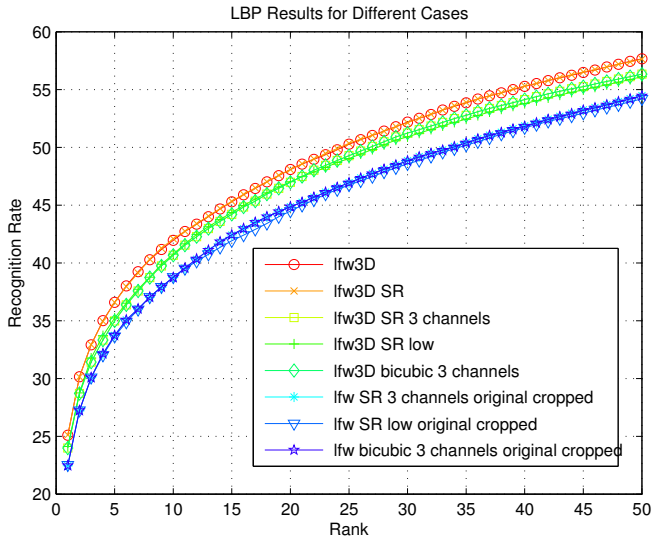
For this, the faces are first frontalized as in [35] and an unsupervised face recognition based on LBP and SURF features are used as a baseline for comparison with the results of proposed experiments (marked as lfw3D in results table and figures). The results of experiment 1 of the bicubic scaling is reported as lfw3D bicubic 3 channels. The super-resolution version is labeled as lfw3D SR 3 channels whereas the super-resolution of the y component is labeled as lfw3D SR low. As for the results of experiment 2, the bicubic scaling is marked as lfw bicubic 3 channels original cropped, and the super-resolution version is labeled as lfw SR 3 channels original cropped. Similarly, for the method of applying SR on the y component only, the results are labeled as lfw SR low original cropped

For experiment 3, face images are processed with the SRCNN algorithm after frontalization to observe the effect of super-resolution on the extracted features. These results are marked as lfw3D SR.

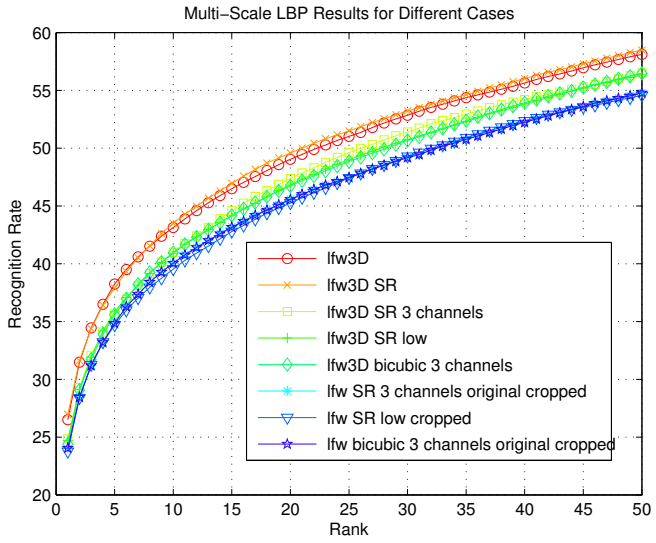
As shown in Figures 4.8 and 4.9, the super-resolution algorithm enhances the recognition rates for both LBP and Multi-Scale LBP features over bicubic scaled version in both experiments. However, both are still behind the baseline recognition rate. Moreover, the recognition rate of experiment 1 is better than the one obtained from experiment 2, which indicates that applying face frontalization before scaling and sharpening process is superior to scaling all the image up then frontalize the detected face. Furthermore, it can be observed that Multi-Scale versions of both LBP and SURF perform better in all experiments outperforming all other features used for this unsupervised test.

As for the results of experiment 3, the super-resolution algorithm positively affect frontalized faces and improves the recognition rate for both types of features in the single and multi-scale versions. Moreover, the SURF features results 4.9, indicate

that the rescaled version of face images after frontalization have performance over the original faces without scaling indicating that a slight blurring of the face image can improve recognition using SURF features.



(a)



(b)

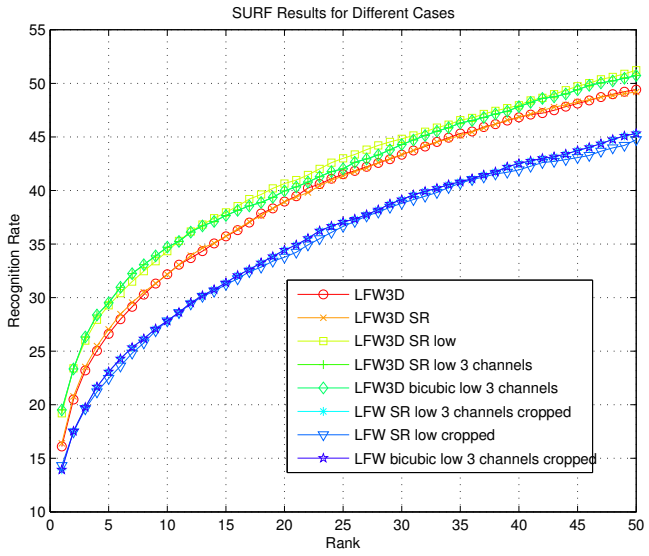
Figure 4.8: Average percentage recognition rate results for both a)LBP b)Multi-scale LBP.

	SURF	Multi-Scale SURF	LBP	Multi-Scale LBP
lfw3D	16.10	24.60	25.09	26.51
lfw3D SR	16.59	24.87	25.32	27.17
lfw3D SR 3 channels	19.72	26.40	24.19	25.03
lfw3D SR low	19.21	25.91	24.12	24.71
lfw3D bicubic 3 channels	19.52	26.20	23.99	24.38
lfw SR 3 channels orig.	14.32	20.54	22.65	24.23
lfw SR low	14.13	20.03	22.51	23.76
lfw bicubic 3 channels orig.	13.93	20.34	22.46	24.04

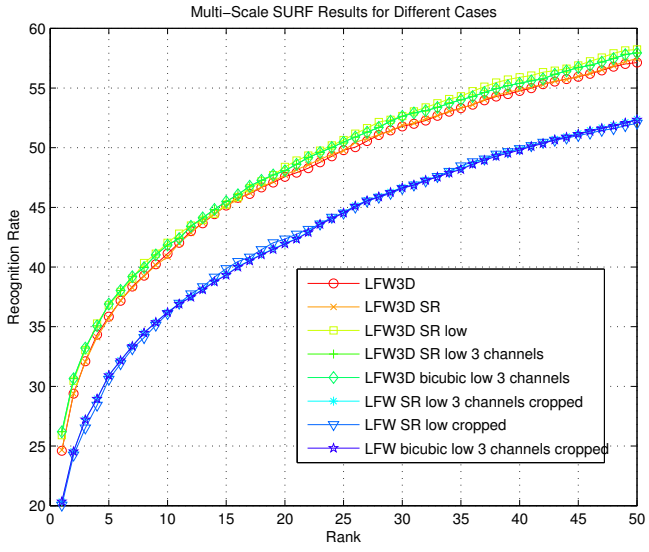
Table 4.3: Average rank 1 recognition rate of all cases in the experiments.

4.6 Conclusions

In this work, an unsupervised face recognition has been applied to the images in the labeled faces in the wild dataset with LBP, Multi-Scale LBP, U-SURF and Multi-Scale U-SURF based extracted features. The results indicate that Multi-Scale versions of both features outperform single scale version of the same features and HighDimLBP features with reasonable extraction and distance calculation time. Three experiments have also been introduced to measure the performance of applying single image super-resolution algorithm on faces captured in the wild, and the effect of order of applying it with face frontalization algorithm. It can be concluded that applying super resolution on frontalized faces provides better results compared to other techniques where super resolution is applied first. This is because, similar to bicubic scaling face, frontalization uses interpolation to calculate some pixels values which can be enhanced with super-resolution techniques. The results also indicate that applying super-resolution on bicubic scaled faces shows slight enhancement in unsupervised face recognition process for both experiments for the two types of features.



(a)



(b)

Figure 4.9: Average percentage recognition rate results for both a)SURF b)Multi-scale SURF.

CHAPTER 5: UNSUPERVISED SUB-GRAPH SELECTION AND ITS APPLICATION IN FACE RECOGNITION TECHNIQUES

5.1 Introduction

As mentioned in Chapter 2, dataset partitioning technique has been used to improve face recognition rate, even though this technique has limitations due to the unconstrained nature of the dataset. With this motivation, an unsupervised grouping technique is proposed in this chapter to address this issues by grouping images without considering the identity of the person. This work distinguishes itself from its counterparts and contributes to the related literature by:

(1) Introducing a new unsupervised grouping technique for large training datasets,

(2) Applying different grouping criteria in the proposed method,

(3) Demonstrating the efficiency of the proposed method by providing a comparative study using multiple databases.

The chapter is divided into five sections. The following section, Section 2, explains the sub-graph selection process. This is followed by a comprehensive description of the proposed hierarchical algorithm (Section 3). Section 4 demonstrates how to further improve the recognition rate by optimizing the grouping process. Section 5 depicts

the results of the proposed technique. Conclusions and future work are discussed in Section 6.

5.2 Sub-Graph Selection Process

The sup-graph selection process requires selecting a sub-graph k_o from a graph G with a specific criterion. The algorithm assumes that all training face images are a fully connected graph (G) with number of nodes (L) and the edge between every two nodes w_{ij} is the sum of the Euclidean distances between the features of these the nodes i and j. The goal is to obtain the best sub-graph set $S = \{k_1, k_2, k_3, ..., k_N\}$, where each sup-graph has a number of nodes (l), k_o is the sub-graph number o, and N is the total number of reconstructed sub-graphs that will be used in the hierarchical technique. Different strategies for this sub-graph selection process are investigated including 1) minimizing the weight of the sum of edges within the entire sub-graph; 2) randomly choosing nodes for the sub-graph and, 3) maximizing the weight of the sum of edges within the entire sub-graph. After sub-graphs are created, regular face recognition technique (Eigenface, in this case) is applied to each fully connected sub-graph to select the top best matches from each group. These sets of matches from the first sub-graph set form the subsequent level of sub-graphs. This process is repeated until a single small fully connected graph of (l) nodes remain. This hierarchical grouping algorithm is presented and different variations are explored by testing it on benchmark datasets to prove the possible improvement in the recognition rate over fully connected images graph. Figure 5.1 shows 2D example for different strategies for the sub-graph selection process.

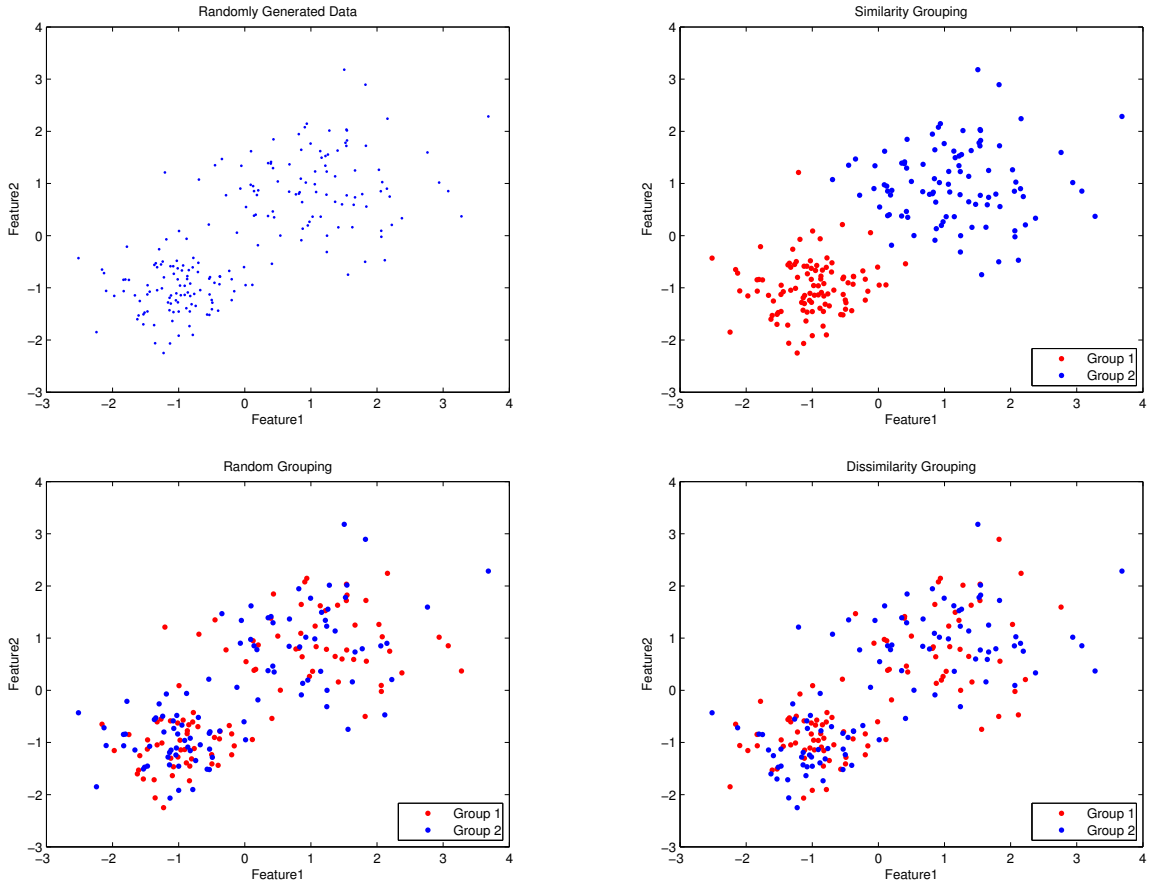


Figure 5.1: 2D example for two sub-graphs selection.

5.3 Face Recognition using Hierarchical Sub-Graph Selection (HSGS) Algorithm

Testing various face recognition algorithms proved that rank 1 recognition rate in unsupervised algorithms drops down if the number of images in the dataset is approximately above some threshold value (e.g. hundred in case of standard Eigenface technique). In order to understand the impact of smaller subsets generated from the entire training set, this work proposes the following. As also detailed in the previous sections, assuming that the face images are a fully connected graph (G) with number of nodes (L) with a goal to select the best sub-graphs set $S = \{k_1, k_2, k_3, ..., k_N\}$each

having a number of nodes ($l \leq 100$), where N is the number of reconstructed sub-graphs to improve the recognition rate over the hierarchical technique. With this goal, applying recognition algorithm over each of these sub-graph sets (groups), a few top matched nodes from each sub-graph (group) (2 to 5) are selected. New groups are then generated from these top matches. Depending on the number of images in the dataset, a number of hierarchical levels are created. Recognition algorithm (e.g. Eigenface) is then applied on each level group. As the final step, the top matches from the final subgroups are collected, and recognition algorithm is re-applied on this final group to select the best-matched image. Figure 5.3 shows the block diagram of the proposed hierarchical technique with the Eigenface as an example of recognition method. The main challenge of this technique is to determine the best sub-graphs selection strategy to improve the overall face recognition rate. There are three possible grouping strategies: i) *Similarity* Grouping by minimizing the sum of weights in the entire sub-graph where similar images are added to the same group (the similarity measurement is the distance between faces features, e.g. pixels gray level). This can be achieved by using regular clustering techniques, ii) *Random* Grouping by assigning the images to the groups (sub-graphs) randomly, iii) *Dissimilar* Grouping by maximizing sum of weights in the entire sub-graph, in other words, maximizing the standard deviation within the same group where the grouping process based on dissimilarity (Maximizing metric distance between faces features in the same group). An additional challenge is to obtain the suitable number of levels in the hierarchical system along with the number of matched images to be selected from each level to feed into the next level in the hierarchy. In order to achieve this, these three possibilities have been tested on a large dataset, viz., Extended Yale B+ [52], FERET [53][54] and FRGC v2 [55], which having different positioning and illumination levels to determine the best approach for the hierarchical face recognition technique. Figure 5.2 shows

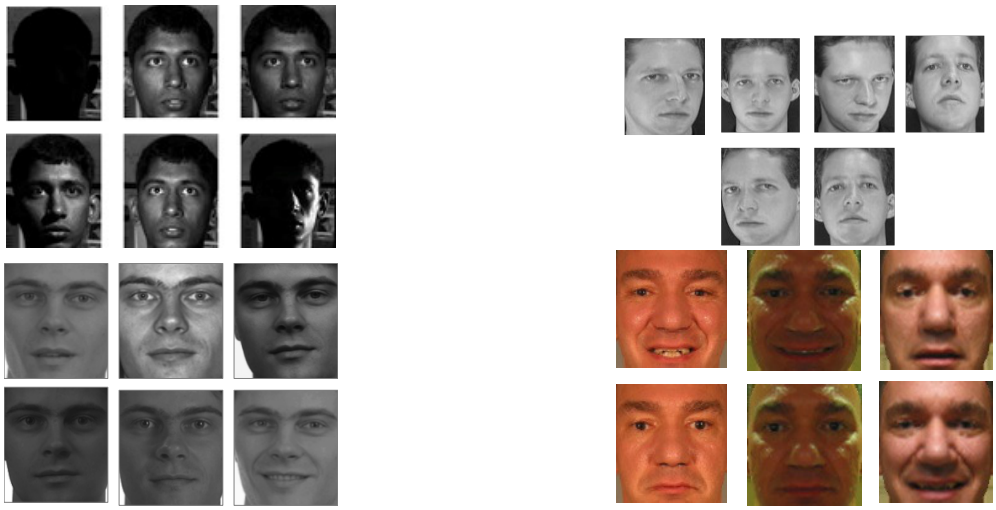


Figure 5.2: Examples of the dataset images used.

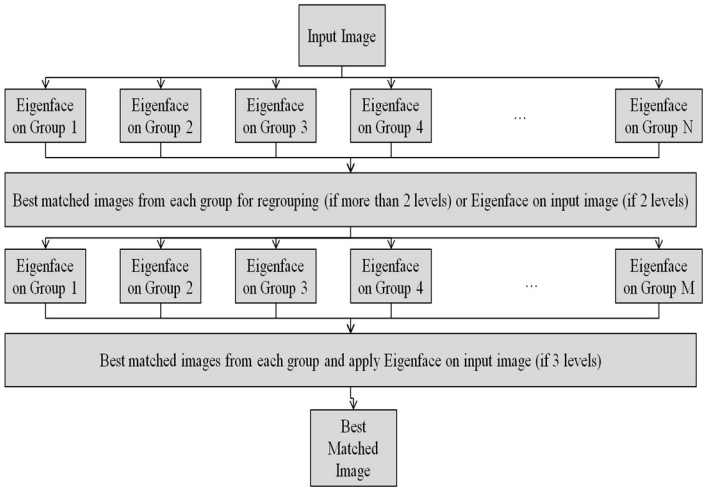


Figure 5.3: The proposed hierarchical system for rank 1 recognition

examples of datasets images for the proposed sub-graph selection algorithm.

5.3.1 Optimized Dissimilarity Sub-Graph Selection Technique

As also detailed in the results section, the simple dissimilarity measurement, viz. maximizing distances between in group images by taking the mean image as a reference, performed superior compared to the other two grouping techniques, i.e., similarity and random selection. However, this method suffers from drawbacks since these

criteria will not guarantee the exact dissimilarity between each group's images. To explain further, consider a 2-D set of (x,y) points. If the training dataset includes {(-2,3),(2,3),(-2,-3),(2,-3)} and is required to group these values into two groups based on dissimilarity, then the mean point will be (0,0) and the Euclidean distance between each one of these points and the total mean will be similar for all four points. This will result in poor grouping. It can easily be observed that the best dissimilarity grouping for this case would be {(-2,3), (2,-3)} as one group, and {(2,3), (-2,-3)} as the second group. Mathematically, the variance between all the sub-graph (group) nodes over all basis should be maximized. Therefore applying this method to a face image dataset sub-graphs selection leads to equation (1):

$$\sigma_{total} = \sum_{l=1}^{N} \sum_{k=1}^{m \times n} \sigma_{lk}, \tag{1}$$

where m and n are the number of rows and columns of the face image respectively (assuming that the pixels gray level are the image features), N is the number of extracted sub-graphs. σ_{lk} is the standard deviation of image dimension k in the sub-graph l. Equation (1) will be valid if the number of hierarchical grouping levels is 2. If dataset is very large however, a regrouping is required again to the third or higher levels. To ensure this, an additional term guaranteeing that the variance of the next grouping stage is also be maximized is included in Equation (1). This term deals with the inter-sup-graphs mean (the difference between means of different groups), forcing groups far from each other to have the maximum variance between its group members:

$$\mu_{diff} = \sum_{j=1}^{N} \sum_{i \neq j}^{N} d\left(\mu_i, \mu_j\right), \tag{2}$$

where $d\left(\mu_i, \mu_j\right)$ is the Euclidean distance between the mean of sub-graph i and the mean of sub-graph j. Equation (3) is the required objective function to be maximized:

$$\max_{I_{ij}} g\left(I_{ij}\right) = \max_{I_{ij}} \left(\sigma_{total} + \mu_{diff}\right), \tag{3}$$

where I is the face image vector. Equation (3) can be expressed in terms of minimization as given in equation (4).

$$\min_{I_{ij}} g\left(I_{ij}\right) = \min_{I_{ij}} \left(-\sigma_{total} - \mu_{diff}\right). \tag{4}$$

It has been reported in that L1 (absolute difference), Cosine and χ distance (Chi square) metrics can in some cases work better than L2 (Euclidean distance) to measure the distance between two projected images in the feature space. Therefore, the effect of both metrics (L1 and L2) in the grouping process are tested to obtain the best recognition rate. The optimized group generation using equation (4) can be achieved as a separate process through the utilization of meta heuristics such as Simulated Annealing.

5.4 Results

The results section is divided into four parts for four different datasets. These datasets have benchmark face images to test in several conditions. These dataset are the ORL AT&T, Extended B+ Yale, FERET and FRGC-2 datasets. Each part provides comparisons between the different grouping techniques as well as the difference when using optimum dissimilarity metric. The proposed techniques have been implemented using MATLAB on Ubuntu 14.04 OS.

5.4.1 ORL AT&T dataset

Total number of images in the dataset is 400. These images have been divided into two sets. A set of 200 images for training, and another set of 200 images for testing. Since the number of images in the dataset is not too large, two level hierarchies have

Table 5.1: Rank 1 match of ORL AT&T dataset

Method	Recognition Rate
Original Eigenface [3]	92.5%
Similarity Grouping	94.5%
Random Grouping	94.0%
Mean Dissimilarity Grouping	93.5%
Optimum Dissimilarity L2 Metric	95.0%
Optimum Dissimilarity L1 Metric	94.0%

been implemented. The group size is considered as 50 images for the first grouping level. The following table shows the results obtained for rank 1 match:

5.4.2 Extended Yale B+ dataset

The number of images in the dataset is 14,800. Similar to the ORL AT&T dataset the images are divided into two sets, one with 7,400 images for training, and the other of 7,400 images are for testing. Due to the large number of images, this two-levels recognition did not provide significant improvement in recognition rate. Therefore, a three-levels hierarchy has been utilized. The training images have been divided into 140 groups, each with approximately 50 images. Three different grouping strategies have been tested. Following results are obtained for rank 1 best match:

- The original Eigenface algorithm recognition rate is 55%.
- Similarity Grouping: The recognition rate improves to approximately 77% - 83.5% depending on the number of best images selected from each group in the first level, and the number of groups in the regrouping step in the second level. A recognition rate of 83.5% is achieved when the best 5 matching images are selected from each first level group. A group constructed from these images is regrouped into the next grouping level. Then best 5 matches are selected from each subgroup, and a final Eigenface step is applied on these to obtain the best match. Results are shown in 5.4. The main disadvantage of this grouping method is that, the execution time increases

when the number of groups in the second level increases. The algorithm used for grouping is the K-means clustering algorithm.

• Random Grouping: The recognition rate improves to a range of 88% - 88.65% depending on the number of best matching images selected from the first level groups (from 2 to 5), and the number of groups in the regrouping step in the second level. The recognition rates are noted to be less dependent on the number of best images selected from each group. Further, the execution time is almost independent of the number of best images selected from a group, and the number of regroups.

• Dissimilar Grouping (based on the L2 distance from the mean image on the training set): The recognition rate improves to the range of 89% - 90.15% depending on the number of best images selected from each group in the first level, and the number of groups in the regrouping step in the second level. The recognition rates are less dependent on the number of best images selected from first level groups. Further, the execution time is almost similar for any number of best matches selected from a group, and number of regroups.

• Optimum Dissimilarity (based on L2 metric): Images are grouped based on the stated objective function in three level hierarchy with L2 metric. This method improved rank 1 rate to 91.5%.

• Optimum Dissimilarity (based on L1 metric): Images are grouped based on the stated objective function in three level hierarchy with L1 metric. This method improved rank 1 rate to 93.6%. Also, for this dataset, the probability that the correct person appears in the best top 10 images is tested (rank 10), as shown in 5.5.

The results depicted in 5.4 indicate that the recognition rate increased significantly when the hierarchical technique was used, especially for large databases (Extended Yale B+). The recognition rate has improved further by the proposed optimum dissimilarity grouping criteria. In summary, compared to the results in [56, 57, 58, 59]

Method	Recognition Rate
ICA [56]	82%
Boltzmann Machine [59]	83%
Our approach (Optimum Dissimilarity L2)	91.5%
Our approach (Optimum Dissimilarity L1)	93.6%

Table 5.2: Comparison of Rank 1 Recognition Rate on Extended Yale B+ dataset using different techniques.

where the ICA and Boltzmann machines are used on the same dataset (around 82% for ICA and 83% for Boltzmann approach), the recognition rate of the proposed algorithm is superior.

Based on these results and to simplify testing on other datasets, only optimum dissimilarly with L1 will be used .

5.4.3 Face Recognition Technology (FERET) dataset

For this dataset the fa gallery images are used for the training and groups buildings. The main advantage of this dataset is that it has only one image per person in the gallery or the prob sets, which makes the test more challenging. There are four different sets of prob images (fb, fc, dub1, dub2), which are different in terms of the time of image capture and the light conditions. The four prob sets have been tested against the hierarchical groups implemented from the gallery set. Again, as in the Extended Yale B+, a three level hierarchy has been used since the number of images and classes are 1196, which is considered to be large number of classes. The sub-graphs groups have been built on 26 groups each of them has 52 images. The testing results for rank 1 recognition rate for the different grouping criteria is presented in Figure 5.6. These results show that using the hierarchical recognition with sub-graph grouping improved the rank 1 recognition rate over the regular Eigenface with L2 metric. Also it can easily be observed that the improvement achieved by mean based and optimum

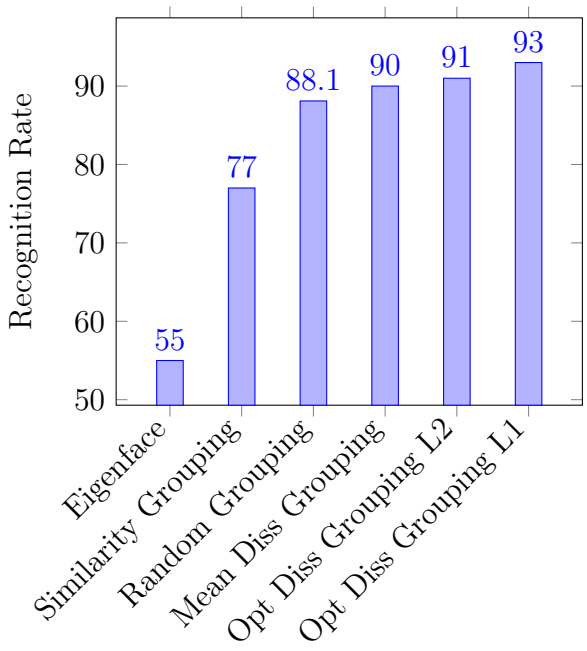


Figure 5.4: Rank 1 recognition rate of different techniques for Extended B+ Yale dataset.

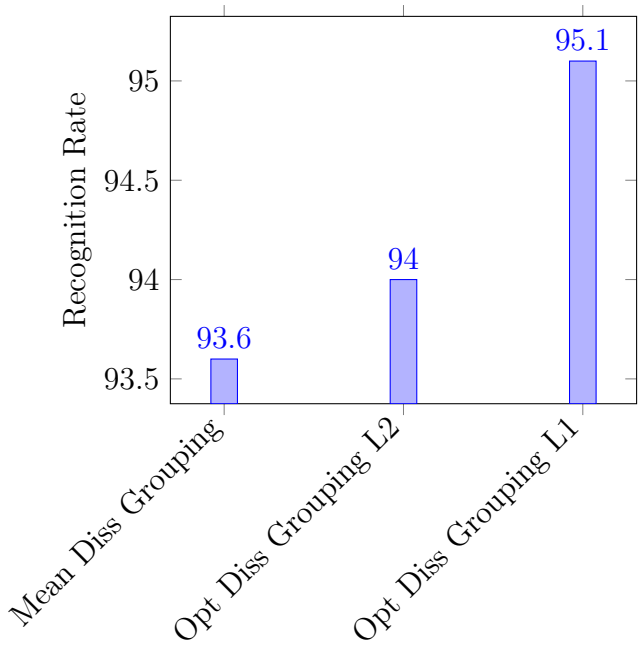


Figure 5.5: Comparison between rank 10 recognition rate of dissimilarity grouping techniques for Extended B+ Yale dataset.

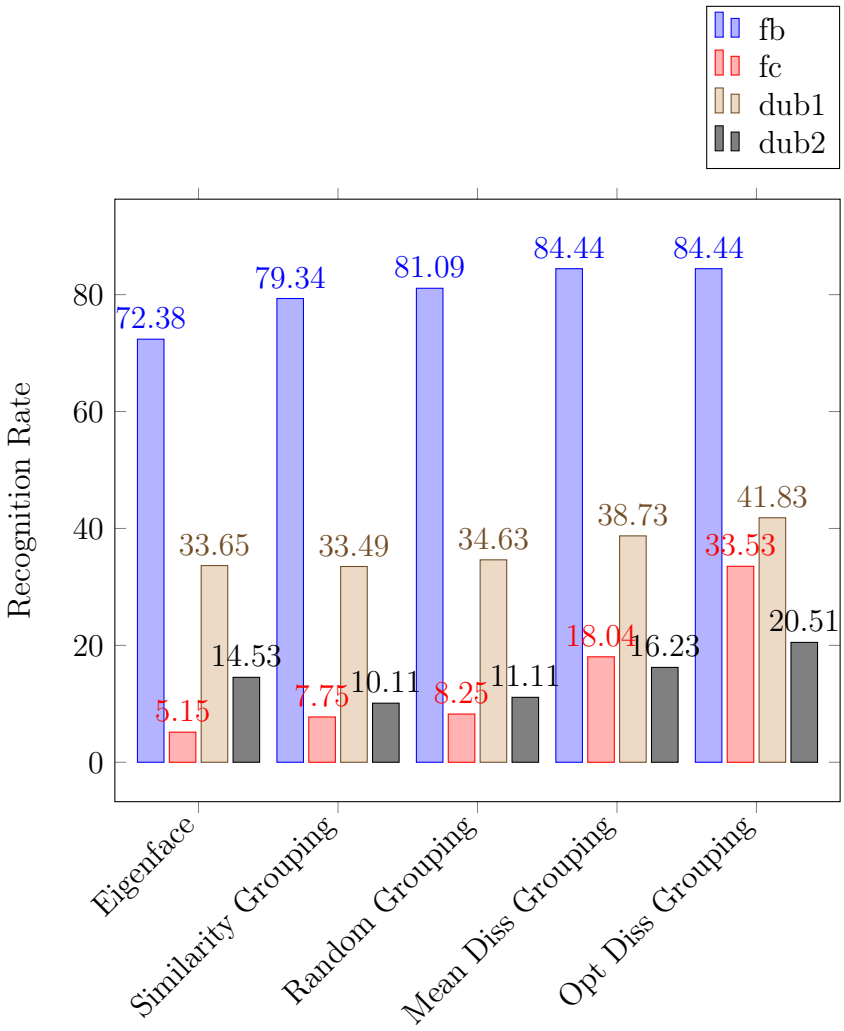


Figure 5.6: Rank 1 recognition rate of different techniques for prob sets of FERET dataset.

dissimilarity outperforms similarity and random grouping techniques. Although the results for this dataset are still below the state-of-the-art results published in [60], but they are still comparable with other techniques such as elastic bunch graph matching (EBGM) [61] and Baysian Map [62], especially for fc, dub1 and dub2 as shown in Table 5.3. Also because this grouping technique is independent of the unsupervised recognition method used, it can also be used with the state-of-the-art features as LBP and SURF, which can contribute to improve these results.

Methods	fb	fc	dub1	dub2
using EBGM [61]	85%	38%	42%	21%
using Baysian Map [62]	82%	34%	45%	29%
Our approach (Optimum Dissimilarity)	84.4%	33.5%	41.8%	20.5%

Table 5.3: Comparison Rank 1 recognition rate results on FERET dataset.

Method	Recognition Rate
using Ahonen's LBP [60]	82.72
using Zhang's LBP [64]	84.17
Our approach (Optimum Dissimilarity)	93.51

Table 5.4: Comparison of different results on FRGC experiment 1 dataset.

5.4.4 Face Recognition Grand Challenge (FRGC v2) dataset

The FRGC v2 dataset is designed with a large number of images per person with high resolution images captured in different years. For this test, experiment 1 is used, where only 2D images are utilized for training and testings. The size of the training set is 12,776 images for 466 persons. The target and the query sets are originally designed for the verification task. Therefore, a closed set recognition protocol is utilized, and the images of the persons not included in the training dataset have been removed. The final number of images for target and query sets are 6,848 images. The training set has been divided into 252 groups using the 4 proposed grouping techniques. The results of the 4 tests compared to the original Eigenface results are shown in Figure 5.7. From the results it can be observed that the mean based and optimum dissimilarity methods outperform the Eigenface and all other grouping criteria. In addition, compared to the state-of-the-art results published in [63], the accomplished results in this research outperforms the recognition rate using some types of LBP on the same experiment for the same database as shown in Table 5.4.

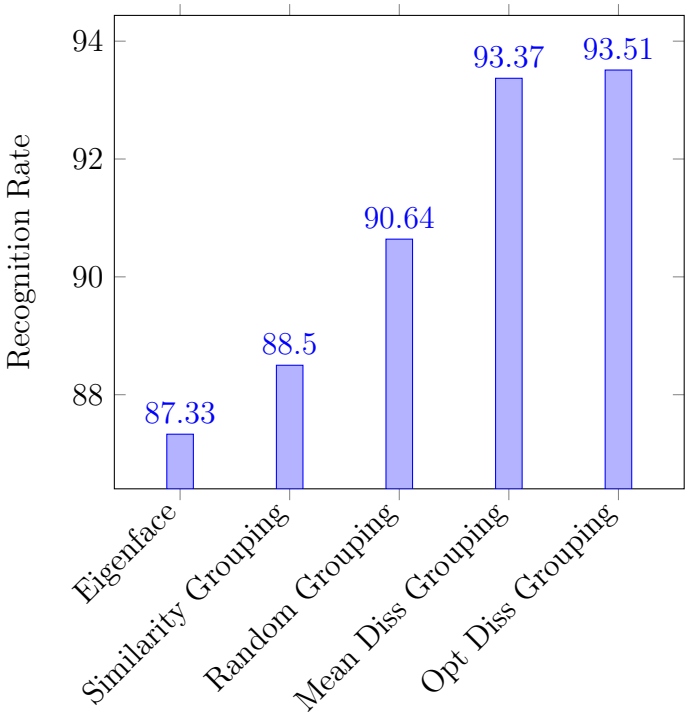


Figure 5.7: Rank 1 recognition rate of different techniques for Experiment 1 of FRGC-v2 dataset.

5.5 Conclusions

This chapter presented a hierarchical sub-graph selection algorithm that aimed at overcoming the limitations brought by large datasets to the standard face recognition algorithms. The algorithm is based on creating small sub-graphs, selecting best matches from each sub-graph, and then dynamically creating next-level sub- graphs until a single group remains. The best match from this final group is accepted as the rank 1 final result of the face recognition process. The study also investigated the best approach for creating sub-graphs by developing an objective function that can be used for best dissimilarity between groups at all levels. Detailed testing on large benchmark datasets indicated that the proposed method produced the best results with a sub-graph size of approximately 50 nodes (images) for the Eigenface technique.

Compared to the standard Eigenface algorithm, the new hierarchical sub-graph selection algorithm improved the recognition rate by more than 40% on some datasets, and with an average by more than 2% over the mean based dissimilarity method. Another advantage of this algorithm is that it uses all the training datasets as one bulk with the unsupervised grouping technique, which is completely independent of the face background and illumination levels. The future work involves applying the hierarchical technique to additional unsupervised face recognition algorithms such as Independent Component Analysis (ICA), KPCA, LBP, SURF and other computer vision algorithms that suffer from degradation in recognition rate due to large dataset size.

CHAPTER 6: NEURAL GENERATIVE MODELS FOR 3D FACES WITH APPLICATION IN 3D TEXTURE FREE FACE RECOGNITION

6.1 Introduction

Rapid improvements in 3D capturing techniques increased the utilization of 3D face recognition especially when the regular 2D images fail due to lighting and appearance changes. The techniques used for 3D based face recognition have been summarized in [65, 66, 67]. Relevant studies are explained in the following text.

The work in [68] uses 3D face recognition by segmenting a range image based on principal curvature and finding a plane of bilateral symmetry through the face. This plane is used for pose normalization. The authors consider methods of matching the profile from the plane of symmetry and of matching the face surface. A modified technique proposed in [69], where the authors use segment convex regions in the range image based on the sign of the mean and Gaussian curvatures, and create an Extended Gaussian Image (EGI) for each convex region. A match between a region in a probe image and in a gallery image is done by correlating EGIs. A graph matching algorithm incorporating relational constraints is used to establish an overall match of probe image to gallery image. Convex regions are believed to change shape less than other regions in response to changes in facial expression. This gives

this approach some ability to cope with changes in facial expression. However, EGIs are not sensitive to change in object size, and hence two similarly shaped differently sized faces will not be distinguishable in this representation. In [70] the author begins with a curvature-based segmentation of the face. Then a set of features are extracted that describe both curvature and metric size properties of the face. Thus each face becomes a point in feature space, and matching is done by a nearest-neighbor match in feature space. It is noted that the values of the features used are generally similar for different images of the same face, "except for the cases with large feature detection error, or variation due to expression" [70]. Instead of working on all face points, in [71] the authors used 3D five feature points only, using these feature points to standardize face pose, and then matching various curves or profiles through the face data. Experiments are performed for sixteen subjects, with ten images per subject. The best recognition rates are found using vertical profile curves that pass through the central portion of the face. Computational requirements were apparently regarded as severe at the time this work was performed, as the authors note that "using the whole facial data may not be feasible considering the large computation and hardware capacity needed" [71]. In [72] they extend Eigenface and hidden Markov model approaches used for 2D face recognition to work with range images. [73] also perform curvature-based segmentation and represent the face using an Extended Gaussian Image (EGI). Recognition is then performed using a spherical correlation of the EGIs. In [74] the authors report on a method of 3D face recognition that uses an extension of the Hausdorff distance matching. Again, work in [75] explores principal component analysis (PCA) style approaches using different numbers of eigenvectors and image sizes. The image data set used has 6 different facial expressions for each of the 37 subjects. The performance figures reported resulting from using multiple images per subject in the gallery. This effectively gives the probe image more chances to make a

correct match, and is known to raise the recognition rate relative to having a single sample per subject in the gallery [76]. Registration and correspondence has been used in [77] to perform 3D face recognition using iterative closest point (ICP) matching of face surfaces. Even though most of the work covered here used 3D shape acquired through a structured-light sensor, this work uses a stereo-based system. The Approach used in [78] is 3D face recognition by first performing a segmentation based on Gaussian curvature and then creating a feature vector based on the segmented regions. The authors report results on a dataset of 420 face meshes representing 60 different persons, with some sampling of different expressions and poses for each person. Another research is perform 3D face recognition by locating the nose tip, and then forming a feature vector based on contours along the face at a sequence of depth values [79]. An isometric transformation approach has been used in [80] to analyze 3D face in an attempt to better cope with variation due to facial expression. Rather than performing recognition on the all face as one module, the authors in [81] have performed recognition using registration on separate face parts and uses fusion to come up with a final decision. Moreover, other research as in [82] and [83] use the high dimensional extracted features, viz. scale invariant feature transform (SIFT, mesh-SIFT) and histograms for both gradient and shapes, from the 3D cloud, and perform the recognition process on them.

As a result of this survey, it can be noted that most if not all research working to extract some features from given 3D face points cloud and use these features in the recognition process. The extracted features are depending directly on the cloud space and can be easily affected by the structure and size of the given points cloud. It is pointed out that these clouds can be modeled to save the storage size or to regenerate the depth information. Other research converts the given 3D points cloud to 2.5D at standard X and Y coordinates using orthographic projection and converting the

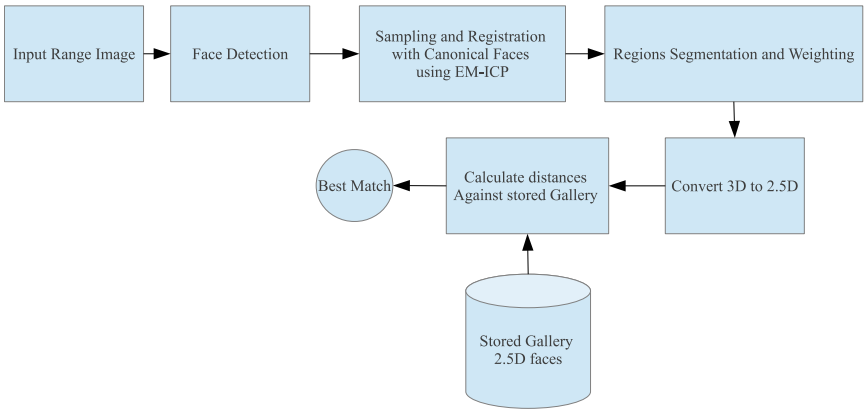


Figure 6.1: Example for 3D recognition using registration and projection to 2.5D.

problem to pixel image recognition [84]. Figure 6.1 shows a complete system that uses this technique.

As previously mentioned, there is literature that focus on modeling the 3D face clouds as in [85]. Here, a neural model is designed as shown in Figure 6.2. In this network, the input consists of second order values for all input points cloud and the output is 0. Additional input values are added for extra generated surfaces inside and outside the point cloud surface. The output in this case should be proportional to the distance from the input point to the cloud surface as shown in Figure 6.3. The model however requires the generation of at least 5 times the number of the original points cloud. Furthermore, the output is not guaranteed to be on the original surface since the acceptance tolerance d is defined for scaling purposes making this model computationally expensive if a higher accuracy required.

Despite the fact that there is an increase in the literature that includes Deep-learning and Deep-neural systems in 3D object recognition as in [86, 87, 88], none of these techniques have been applied to 3D face recognition. One reason for this lack of applicability is the high sensitivity and the closeness of features between the faces of different individuals, specifically if the 3D data is used alone without any texture.

Under this motivation, the main contributions of the proposed 3D neural recognition

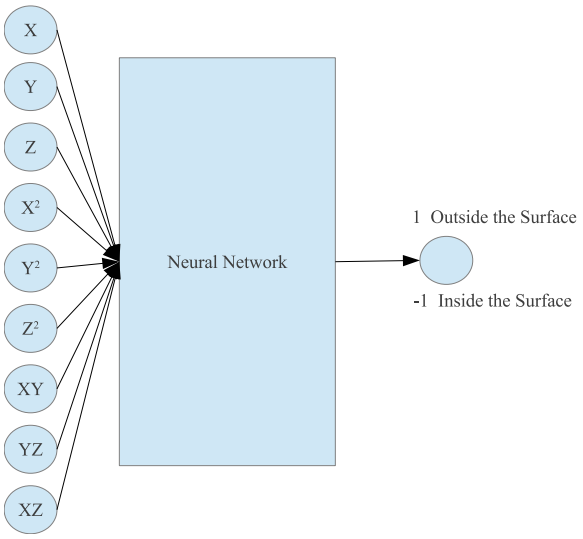


Figure 6.2: Neural network for 3D points modeling in Cretu et al.

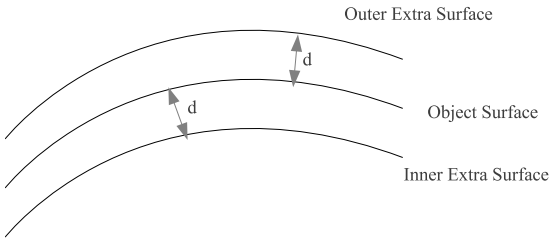


Figure 6.3: Extra surfaces generated for neural model learning in Cretu et al.

and verification system are listed as follows.

(1) Designing neural generative model for representation and reconstruction of 3D faces,

(2) Significant reduction in the storage space used for the 3D point clouds, by replacing the stored point clouds with the generated neural model representations,

(3) Using the generated presentation from the 3D regression models of gallery set for recognition and verification against generated model representation for probe points cloud,

(4) Combining generated face model representation with Siamese network to generate a comprehensive framework for 3D face verification.

6.2 Proposed 3D Based Face Recognition System

In some cases, due to lighting conditions and/or makeup or other 2D effects in the face image, the regular 2D image becomes insufficient for face recognition. In order to address these issues, this work presents a 3D based face recognition system that is able to work on the texture free 3D point clouds extracted by depth cameras to identify or verify the person. In this regard, this research introduces a new technique for 3D cloud regeneration using a neural generative model to handle the differences caused by heterogeneous depth cameras, and to generate a new face canonical compact representation. The proposed system reduces the stored 3D dataset size and if required, provides an accurate dataset regeneration. Furthermore, the system generates neural models for all gallery point clouds and stores these models to represent these faces in the recognition or verification process. For the probe cloud to be verified, the system obtains the 3D points cloud as an input with face landmark points. These landmark points are then registered to reference points to align and to scale the input cloud correctly. After the registration step, a neural model is generated for this prob cloud to provide a compact representation of the 3D face data. The extracted neural model is then applied to a face recognition or verification step to detect the best matched model from the pre-stored gallery model presentations. This work also introduces the utilization of Siames deep neural network in 3D face verification using generated model representation as a raw data for the deep network. The complete proposed system is depicted in Figure 6.4. The following sections will explain each step used in the system.

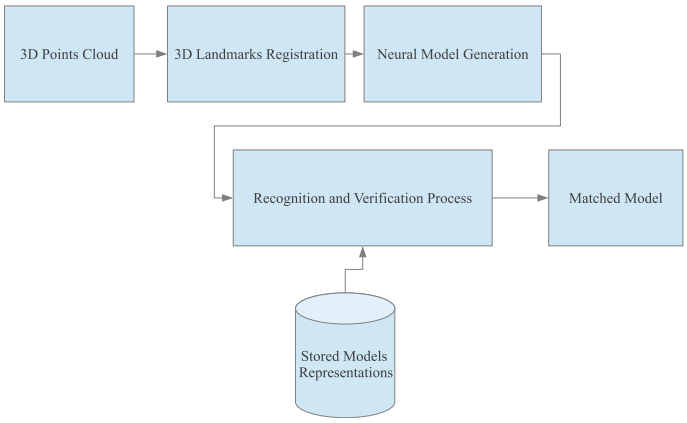


Figure 6.4: Proposed 3D based face recognition system.

6.3 3D Registration

The first step in the proposed system is the registration that transforms all 3D data into a canonical standard position. This step involves 3D data using Iterative Closest Point (ICP) algorithm. Before applying ICP on the input clouds, these clouds have to be normalized around their means and to be placed inside a cube with maximum dimensions of $2 \times 2 \times 2$ with each axis having a range $[-1, 1]$. After the normalization step, the landmark points registration is applied to obtain the suitable rigid transformation between input landmark points and the reference points. The output transformation is then applied to all input points to obtain the registered points which will be used in the following steps.

6.3.1 Iterative Closest Point (ICP)

Iterative Closest Point (ICP), a.k.a., Iterative Corresponding Point, is an algorithm used to obtain the corresponding points and transformations between two groups of points in 2D or 3D. However, the algorithm is mostly used in the 3D cases for registration between some query mesh and standard canonical mesh. The main algorithm was introduced in [89]. In this work the main goal is to obtain the optimum trans-

formation matrix $T = [R|t]$ (where R is the 3×3 rotation matrix and t is 3×1 translation vector) that can convert moving landmark points set $M = \{m_i\}$ to the static points set $S = \{s_i\}$. Assuming that the number of points for both sets are equal ($N_M = N_s$), the objective function required to be minimized should be

$$f(T) = \frac{1}{N_S} \sum_{i=1}^{N_s} \|s_i - Rm_i - t\|^2 . \quad (1)$$

To achieve this goal the following steps are followed:

(1) Find the closest corresponding points (on Euclidean space) between M_k set and S set. These correspondence set will be $Y_k = C(M_k, S)$.

(2) For the particular Y_k, minimize equation (1) solving for the value of T (using least square techniques). The solution will be T_k

(3) Apply $T_k$over M_k set to generate $M_{k+1} = R_k M_k + t_k$

(4) Repeat steps 1,2 and 3 until the stopping criteria are satisfied.

The only problem with this method is its computational complexity which reaches to significant levels when the number of points are large ($O(N_m N_s)$). However, some research uses other techniques rather than Nearest Neighborhood (NN), which used in step 1, to improve the computation. K-D Tree is one of these alternative algorithms that can be used for this improvement. Other modified versions of ICP have also been introduced to make the algorithm more computationally efficient. As also stated in [90] these techniques have been summarized in the following steps:

(1) Selection of some set of points in one or both sets.

(2) Matching these points to samples in the other set.

(3) Weighting the corresponding pairs appropriately.

(4) Rejecting certain pairs by looking at each pair individually or considering the entire set of pairs.

(5) Assigning an error metric based on the point pairs.

(6) Minimizing the error metric.

Based on the new steps, the algorithm will not work on these entire sets, but on some selected samples from both sets. The sampling and rejection steps in the modified algorithm improve the computation complexity, even though this new approach leads to other concerns regarding the best sampling and rejection mechanisms that can be used to obtain the best matching result.

6.4 Neural Regression Model for 3D Face Representation

As mentioned previously, one of the problems in the depth point clouds is the storage size. The storage size for one face representation of about 80,000 points, which is in the order of tens if not hundreds of mega bytes. Therefore, finding an alternative representation that can reduce the size while providing the same accuracy is important. An additional concern is the variability of the number of points from different cameras. This concern is valid regardless of that the image representing a single person or multiple individuals. To address these issues, a new neural representation is proposed as shown in Figure 6.5. The proposed neural model will obtain X and Y coordinates of the input points cloud and will generate the corresponding $\bar{Z}$ values, which should represent the actual Z values of these points. The mathematical representation of the model will be as the following equations:

$$\bar{Z}^i = \tanh\left(n_f^i\right), \tag{2}$$

$$n_f^i = \sum_{j=1}^{M} W_{jf} \cdot \tanh\left(n_j^i\right) + B_f, \tag{3}$$

$$n_j^i = \left(W_{1j} \cdot X^i + W_{2j} \cdot Y^i\right) + B_j, \tag{4}$$

where M is the number of hidden units, (X^i, Y^i, Z^i) is a 3D point in the point cloud and its corresponding output $\bar{Z}^i$ and j is the hidden node index.

To obtain the weights W for a neural model, a loss function should be defined and optimized. In this work, the Mean Squared Error (MSE) will be used as stated in equation (5).

$$\mathcal{L}(P) = \frac{1}{N} \sum_{i=1}^{N} \left\| \bar{Z}^i\left(X^i, Y^i, P\right) - Z^i \right\|^2. \tag{5}$$

where $P = \{W_{1j}, W_{2j}, B_j, W_{jf}, B_f\}$, N is the number of samples in the points cloud, B_j is the bias of the hidden node j and B_f is the bias of the final output node.

To solve this optimization problem, Levenberg–Marquardt Back-Propagation (LM) as discussed in the previous chapter will be used to obtain the values of P that will provide the minimum Mean Squared Error (MSE).

The main advantage of this proposed model is:

- Easer in data augmentation (if the model is used as a raw data features for the verification process as will be explained later): in various machine learning problems, the number of available samples for training or testing are limited. To overcome this issue, data augmentation is commonly used to generate additional samples by manipulating the existing data (via shifting, rotating and clipping in the case of image learning). However, in 3D cases, especially for faces, these processes can generate limited number of samples. For the proposed model however, significantly large numbers of new models can be generated for the same face by only swapping rows in the matrix P. For example, assuming the structure of the network is 2-500-1 (2 is the number of

inputs, M is 500 and 1 output) and the value of B_f will not change and will be added after flatten the matrix, then if $P_1 = \begin{bmatrix} w_{11} & w_{21} & b_1 & w_{1f} \\ w_{12} & w_{22} & b_2 & w_{2f} \\ w_{13} & w_{23} & b_3 & w_{3f} \\ \vdots & \vdots & \vdots & \vdots \\ w_{1M} & w_{2M} & b_M & w_{mf} \end{bmatrix}$ then $P_2 = \begin{bmatrix} w_{13} & w_{23} & b_3 & w_{3f} \\ w_{12} & w_{22} & b_2 & w_{2f} \\ w_{11} & w_{21} & b_1 & w_{1f} \\ \vdots & \vdots & \vdots & \vdots \\ w_{1M} & w_{2M} & b_M & w_{mf} \end{bmatrix}$ will also generate a new model that represents the same face as P_1 by only swapping rows 1 and 3. Therefore, assuming that M equal to 500, this technique can generate 500! different model for the same face from a single model.

- Reduce the storage size of the face representation: because to store a 3D face only the network weights need to be stored (assuming the network structure is known). For instance, if the network structure is (2-500-1), the number of stored weights will be 2001 (taking into account the network biases). This means that instead of storing about 80,000 double precision numbers or more (the original number of points in the cloud), only 2,000 float precision numbers can be stored which will improve the storage size with a factor of 80.

- Can be used in 3D super-resolution: the neural generated model designed in this work is a regression model that can be used for smoother accurate interpolation to generate higher resolution version from the original 3D points, which can considered as 3D super-resolution algorithm.

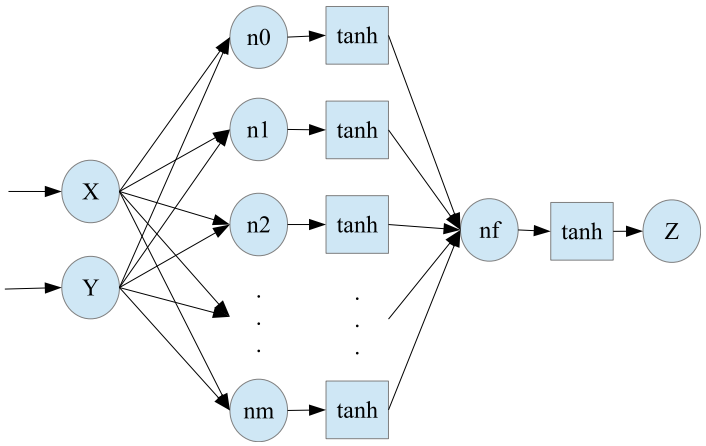


Figure 6.5: Proposed neural representation of face depth data .

6.5 Levenberg–Marquardt Back-Propagation

As shown in Chapter 2, in the Gradient Decent Back-Propagation algorithm, the back-propagated recurrent form of sensitivity at layer k in the neural network has been formulated as

$$\delta^k = \dot{F}^k\left(n^k\right).W^{k+1^T}.\delta^{k+1}, \tag{6}$$

where

$$\dot{F}^k\left(n^k\right) = \begin{bmatrix} \dot{f}^k\left(n^k\left(1\right)\right) & 0 & \cdots & 0 \\ 0 & \dot{f}^k\left(n^k\left(2\right)\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{f}^k\left(n^k\left(S_k\right)\right) \end{bmatrix}, \tag{7}$$

and

$$\dot{f}^k\left(n^k\left(i\right)\right) = \frac{df^k\left(n^k\left(i\right)\right)}{dn^k\left(i\right)}. \tag{8}$$

where $f^k\left(n^k\left(i\right)\right)$ is the activation function of node i in layer k and $n^k\left(i\right)$ is the

output of node i in layer k.

The Gradient Decent method works well when the network task is a classification with softmax loss function. However, for some tasks, the loss function stated in Chapter 2 section 3 is defined as Mean Squared Error (MSE) function between neural network output $a^M$and required output y. Based on this, the loss function can be defined as

$$e_i(W) = \left(y_i - a_i^M(W)\right), \tag{9}$$

$$\mathcal{L}(W) = \frac{1}{2}\sum_{i=1}^{N} e_i^2(W), \tag{10}$$

where $N = Q \times S_m$

Based on Levenberg–Marquardt algorithm used in [33], weights and biases update can be calculated as

$$\Delta W = -\left[\nabla^2 \mathcal{L}(W)\right]^{-1} \nabla \mathcal{L}(W), \tag{11}$$

where $\nabla^2 \mathcal{L}(W)$ is the Hessian matrix and $\nabla \mathcal{L}(W)$is the gradient. The gradient term can be expressed as

$$\nabla \mathcal{L}(W) = J^T(W)\, e(W), \tag{12}$$

where $e(w) = \begin{bmatrix} e_1(W) \\ e_2(W) \\ \vdots \\ e_N(W) \end{bmatrix}$, and the Hessian matrix can be approximated as

$$\nabla^2 \mathcal{L}(W) = J^T(W)\, J(W), \tag{13}$$

where J is the Jacobian matrix stated as

$$J(W) = \begin{bmatrix} \frac{\partial e_1(W)}{\partial W_1} & \frac{\partial e_1(W)}{\partial W_2} & \cdots & \frac{\partial e_1(W)}{\partial W_n} \\ \frac{\partial e_2(W)}{\partial W_1} & \frac{\partial e_2(W)}{\partial W_2} & \cdots & \frac{\partial e_2(W)}{\partial W_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(W)}{\partial W_1} & \frac{\partial e_N(W)}{\partial W_2} & \cdots & \frac{\partial e_N(W)}{\partial W_n} \end{bmatrix}, \tag{14}$$

From equations (12), (13), (14) and (11) can be expressed as

$$\Delta W = \left[J^T(W) J(W)\right]^{-1} J^T(W) e(W), \tag{15}$$

Adding one more control parameter the update can be stated as

$$\Delta W = \left[J^T(W) J(W) + \mu I\right]^{-1} J^T(W) e(W), \tag{16}$$

The new added parameter μ is used as a variable step control for the updates based on the loss value. Every time the loss $\mathcal{L}(W)$ reduces, the value of μ is divided over some other constant parameter β to go closer to the minimum loss value.

Applying the new equation to back-propagation algorithm in section 3 resulting in new weight update equation

$$\Delta w^{k+1}(i,j) = -\alpha . \frac{\partial L_q}{\partial w^{k+1}(i,j)} = -\alpha . \frac{\partial \sum_{m=1}^{S_M} e_q^2(m)}{\partial w^{k+1}(i,j)}, \tag{17}$$

$$\Delta b^{k+1}(i) = -\alpha . \frac{\partial L_q}{\partial b^{k+1}(i)} = \Delta b^{k+1}(i) = -\alpha . \frac{\partial \sum_{m=1}^{S_M} e_q^2(m)}{\partial b^{k+1}(i)}, \tag{18}$$

Identical steps used in the regular back-propagation can also be used with the LM method to fill the Jacobian matrix with small modification at the final step

$$\delta^M = -\dot{F}^M\left(n^M\right). \tag{19}$$

Based on these equations, the LM back-propagation algorithm will work as follows:

Algorithm 2 Levenberg–Marquardt algorithm

Apply all Q inputs to the network and calculate network outputs and errors corresponding to these output and loss value.

(1) Use equations (19), (8), (7), (6), (14) to calculate Jacobian matrix (other efficient Jacobian calculation methods can be used in this step).

(2) Solve equation (16) (using Cholesky factorization) for the ΔW.

(3) Use the calculated ΔW to calculate the new value of $W + \Delta W$.

(4) Check the new loss value, if the loss value decrease, then decrease μ by β , update $W = W + \Delta W$ and go to step 1. If the loss didn't reduce increase μ by β and go to step 3.

(5) Repeat the these steps until the stopping criteria are satisfied.

6.6 Recognition and Verification

Using similarity metric for face verification is proven to be an efficient method, especially if the number of images per class is low. Therefore, this work utilizes the similarity metric method with Convolutional Neural Network (CNN) to perform a Siamese Network that will be applied in the final step of the proposed system to perform the verification process.

As explained in Chapter 3, using Siamese Network for face verification has been introduced in [36], where two input images X_1 and X_2 are applied to the same nonlinear mapping G_W to extract the main features that minimize the main energy function E when X_1 and X_2 belong to the same person and maximize it when they belong to different persons. The typical structure for this network is shown in Figure 6.6 [36]. The formal definition of the function E can be expressed as in equation (20)

$$E\left(W, X_1, X_2\right) = \left\| G_W\left(X_1\right) - G_W\left(X_2\right) \right\|, \tag{20}$$

where W are the shared weight filters between the two input images.

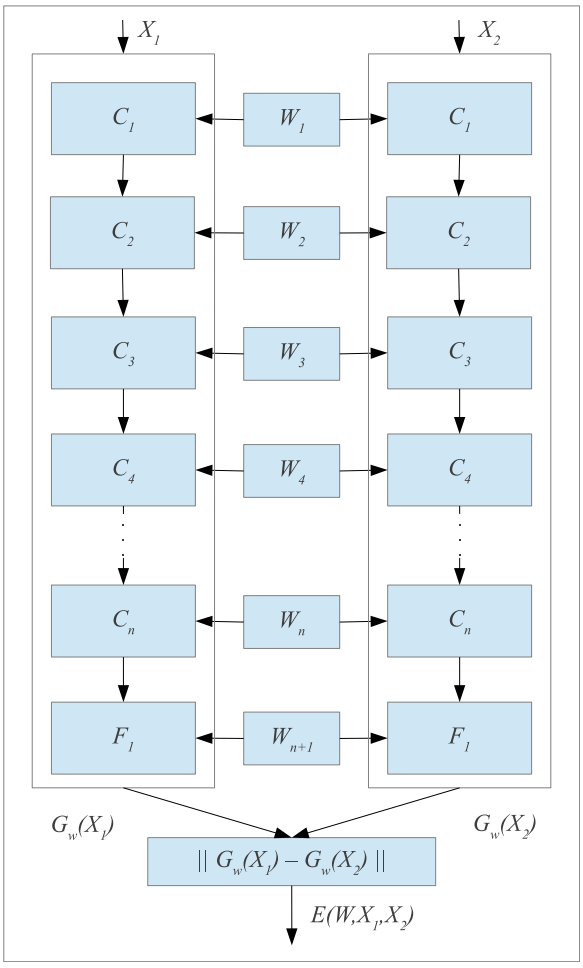


Figure 6.6: Typical structure of Siamese network.

To achieve this goal for the E function, the loss function should monotonically increase with same person pairs' energy and monotonically decrease with different persons pairs' energy. Based on this, the final loss function will be formed as in equations (21),(22),(23),(24),(25)

$$\mathcal{L}(W) = \sum_{i=1}^{N} L\left(W, (Y, X_1, X_2)^i\right), \tag{21}$$

$$L\left(W, (Y, X_1, X_2)^i\right) = Y \cdot L^s\left(E\left(W, X_1, X_2\right)^i\right) + (1 - Y) \cdot L^d\left(E\left(W, X_1, X_2\right)^i\right), \tag{22}$$

$$L^s\left(E\left(W,X_1,X_2\right)^i\right)=\frac{2}{Q}\left(E\left(W,X_1,X_2\right)^i\right)^2, \tag{23}$$

$$L^d\left(E\left(W,X_1,X_2\right)^i\right)=2Q\cdot e^{\left(-\frac{2.77}{Q}E(W,X_1,X_2)^i\right)}, \tag{24}$$

$$Y=\begin{cases}1 & X_1\equiv X_2\\ \\ 0 & X_1\not\equiv X_2\end{cases}. \tag{25}$$

where N is the number of training samples, Y is equal to 1 if X_1 and X_2 belong to the same person and 0 if they present different persons. L^s is the loss function in the case of same persons, L^d is the loss function in the case of different ones and Q is a constant representing the upper bound of E.

Since the energy is monotonically changing for both L^s and L^d, the optimization of the loss function can be easily achieved using simple gradient decent algorithm, and the weights W can be learned using back-propagation algorithm.

Based on that the final step of the proposed system (involves detecting the identity of the query person or verifying his/her identity) will utilize this structure of the network. For the proposed system, the extracted weights P from the previous section constiture the feature vector used in the recognition or verification process. This means that a 1D Siamese structure network will be used for this verification task. The network structure is depicted in Figure 6.6. However, X_1 and X_2 will be replaced by vectorized P_1 and P_2 extracted from the previous step. All calculations stated by equations (20),(21),(22),(23),(24) and (25) will remain identical. However, W will consist of 1D vectors as opposed to 2D as in the original case.

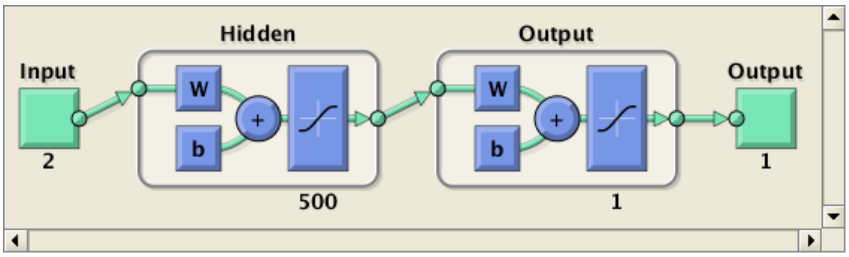


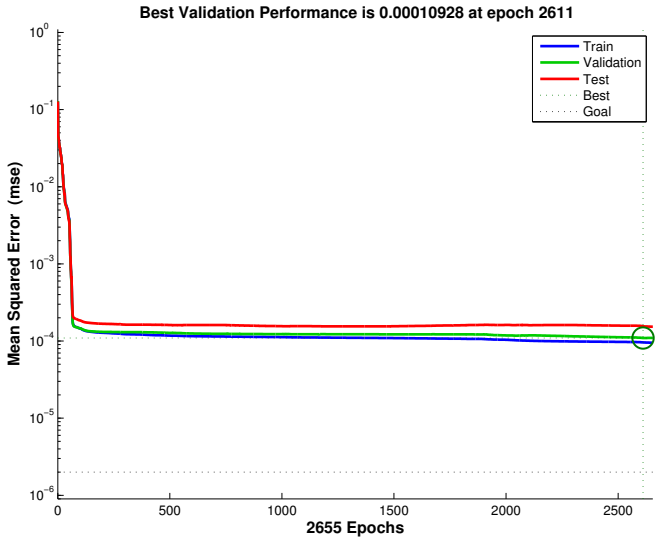
Figure 6.7: Structure of the network used in the experiment.

6.7 Results

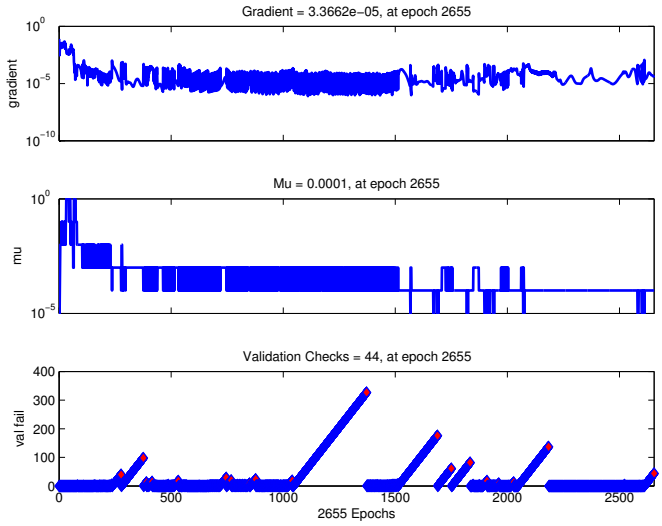
The proposed system has been tested over the Bosphorus [91] database. This dataset consists of 3D and textures for 105 persons with different facial expressions. For this test only the 3D faces with natural expressions are used to test the efficiency of the proposed neural model. Each person has between 1 to 4 natural faces with a total of 299 natural 3D texture free faces. The target mean squared error (MSE) value for the trained models is below 0.0002. The network structure for this problem is provided in Figure 6.7. For the sake of simplicity, only one hidden layer is used in the experiment with the number of nodes in the hidden layer being 500. The proposed neural model has been implemented using Neural Network Toolbox of MATLAB.

As shown in the sample training in Figure 6.8, the training loss improved in the first 100 epochs. Following this, all upcoming epochs worked as fine tuner for the learning parameters.

A sample of the resulting regression model accuracy is shown in Figure 6.9. It can easily be observed that the accuracy of the generated model is more than 99% for presenting target points cloud. Out of all trained models, 100 models achieved the required training MSE of 0.0002, and 170 models terminated when the maximum number of epochs were reached with an average MSE of 0.00031, which is still considered to be very low error. Also as shown in Figure 6.10, only few models (29 models)



(a)



(b)

Figure 6.8: Training performance of one of the generated models.

terminated for achieving maximum gradient or maximum μ value for equation (16). Figure 6.11 shows a sample of the generated points cloud compared to the original one. As also seen in this figure, the original depth points cloud contains a significantly large noise due to camera and environment. However, the generated points cloud is smoother and provides better view of 3D faces.

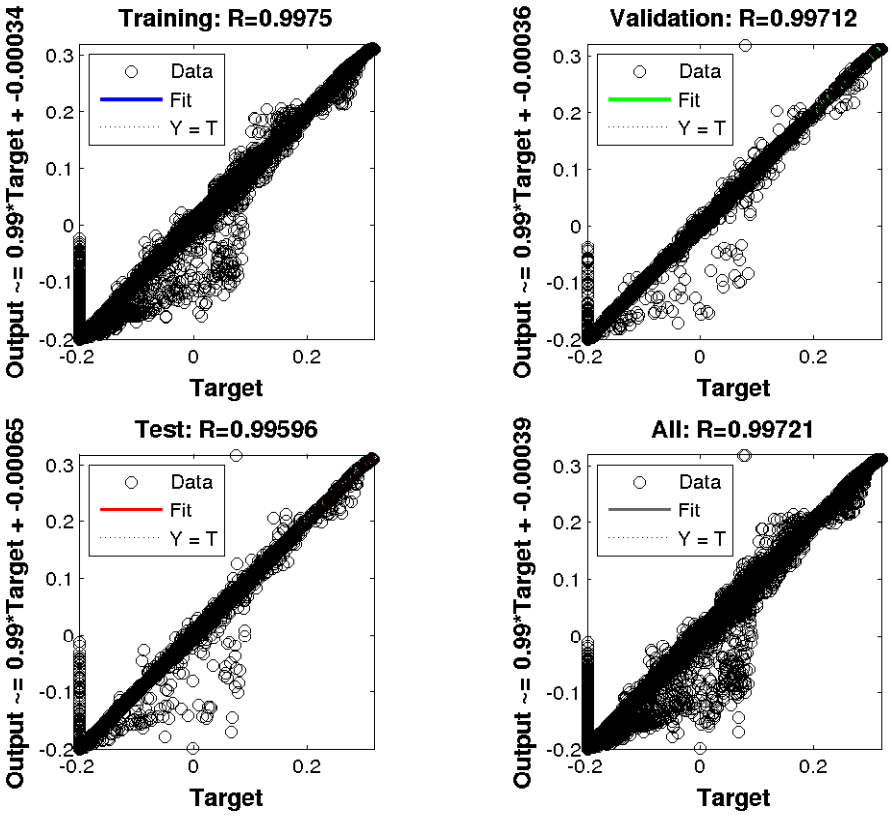


Figure 6.9: Regression result for one of the generated model.

A Siamese network followed by a verification system has been implemented using Caffe library with Python wrapper. The structure of the this network is shown in Figure 6.13. The 299 neural models which are generated for the 105 persons have been used for training and testing. Pairs from the generated models have been selected as a positive (pairs for the same person) and negative (pairs for different persons) training data for the Siamese network. Since the number of positive samples are so limited, the data augmentation technique proposed in section 4 has been used to generate additional pairs for the training and testing. Using this data augmenting technique, the number of generated samples for same person class (positive pairs) is 50,000 samples and for different person class (negative pairs) is 70,000 samples. As it can also be seen from Figure 6.12, the loss function of the trained network did not over-

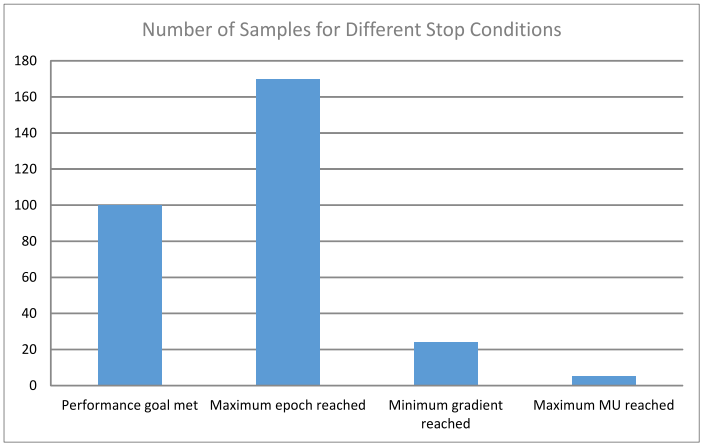
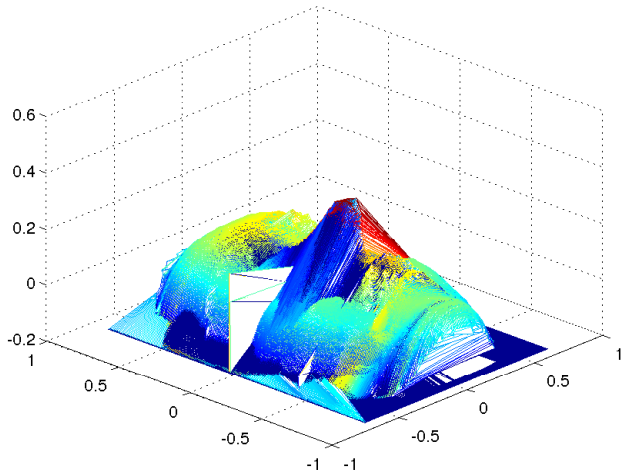


Figure 6.10: Number of training models and their stop conditions.

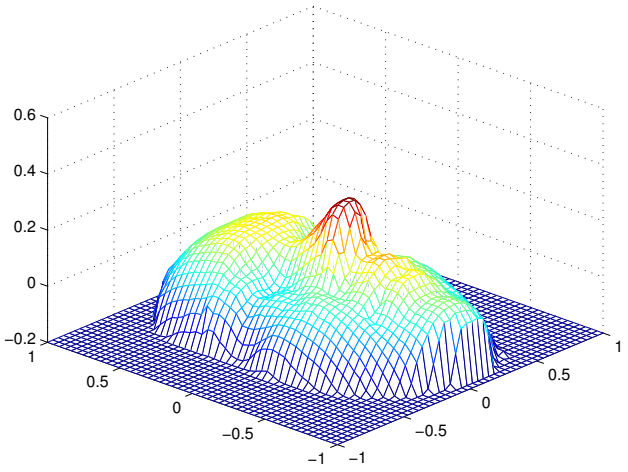
Method	Recognition Rate (%)
ICP-based holistic approach [81]	71.39
Average Regional Models (ARMs) [81]	95.87
HoG+HoS+HoGS [82]	98.8
proposed Siamese Network	98.82

Table 6.1: Comparison of recognition rate using different techniques over 3D faces of Bosporus dataset.

fit, and the network is efficiently trained. The accuracy of the trained network over training pairs achieved %99.9 and %98.82 for testing pairs. The receiver operation characteristic (ROC) and Precision-Recall curves for the trained network over the testing pairs are also shown in Figure 6.14. As also provided in Table 6.1, the achieved verification performance is comparable to the state-of-the-art results published on the same dataset on the natural expression faces. These techniques include ICP, Average Regional Models (ARMs), Histogram of Gradient (HoG), Histogram of Shape index (HoS), Histogram of Gradient Shape index (HoGS) and the fusion of these histograms (HoG+HoS+HoGS). Once again, these generated neural models for these faces can regenerate the 3D point clouds again to be utilized with any of these mentioned recognition technique.



(a)



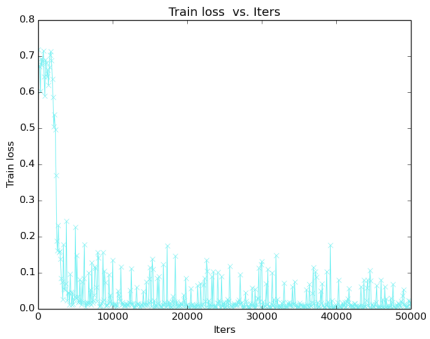
(b)

Figure 6.11: Sample of original depth points cloud (a) and points cloud generated by regression model (b).

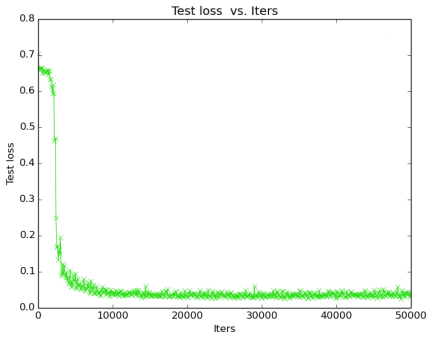
6.8 Conclusions

In this work, a neural generative modeling technique for texture free 3D faces has been proposed. This neural models have been used for presentation and regeneration of the 3D faces. The proposed models have been proved to be an accurate represen-

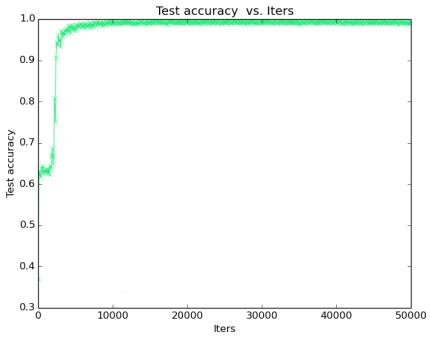
tation of the original 3D point clouds with additional benefits such as reducing the storage size of the 3D cloud, and its ability to accommodate interpolation and 3D super-resolution. The weights of the trained models have been used as a raw data with Siamese CNN network for a complete neural 3D face recognition and verification system. These weights have advantage over the regular 3D clouds for data augmentation. Furthermore, they allow generation of additional models from a given single model, which makes this technique advantageous for small dataset recognition (one of the limitations of using CNN that it required large dataset for training and validation). The Siamese network has been trained over the generated pairs (positive and negative) from the trained models. The results obtained from the trained Siamese network outperformed all reported results over the Bosporus dataset for the natural 3D faces.



(a)



(b)



(c)

Figure 6.12: (a) Training loss of the Siamese Network over 50000 iterations and (b) testing loss of the Siamese Network and (c) testing accuracy of the same network for verification.

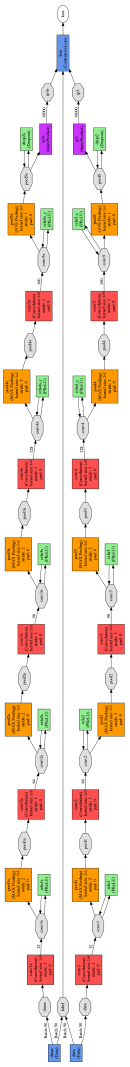
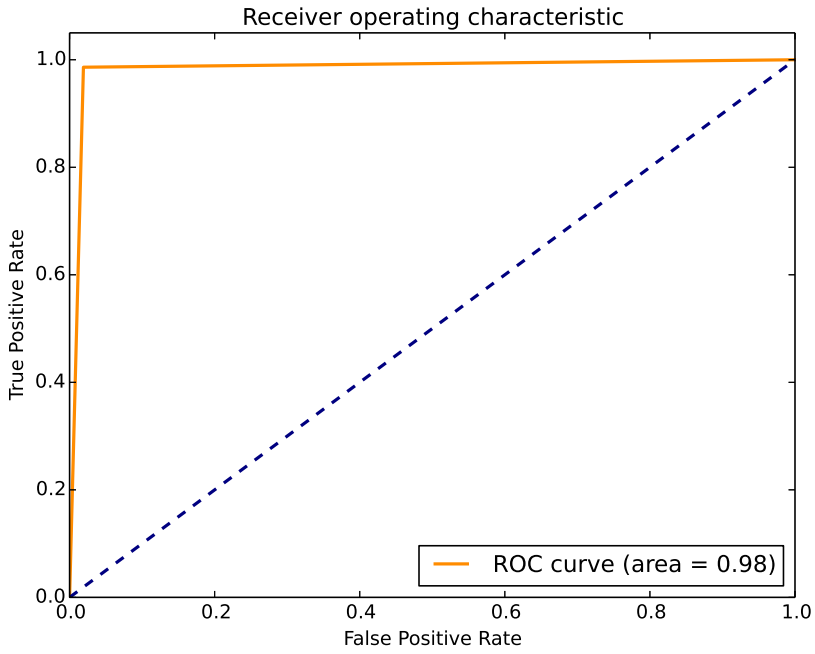
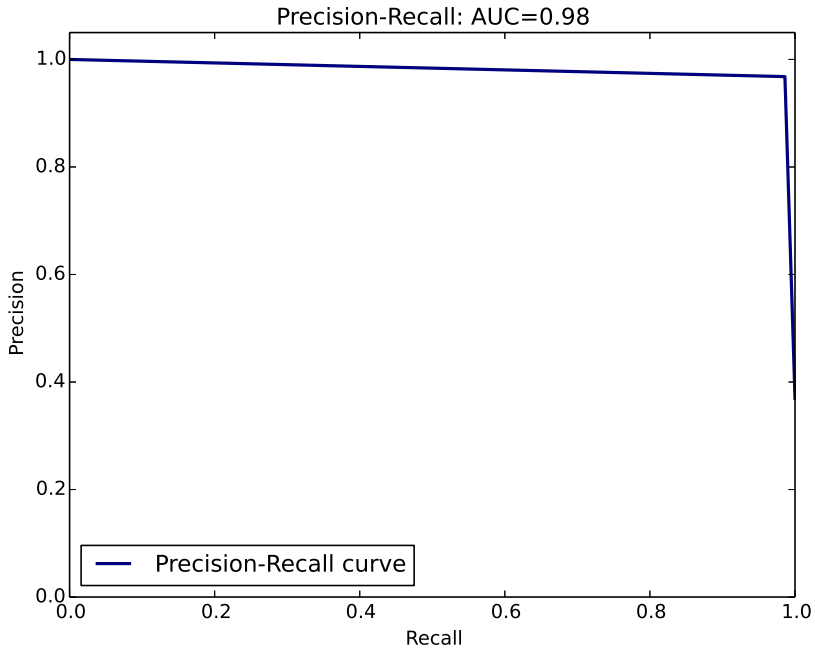


Figure 6.13: Proposed structure of Siamese network



(a)



(b)

Figure 6.14: (a) ROC curve of the Siamese network over testing data and (b) Precision-Recall curve for the same network on the same data.

CHAPTER 7: CONCLUSIONS

This dissertation presented two systems for face recognition. The first one involves a framework for unconstrained 2D large scale face recognition. To handle large scale datasets, a new data manipulation algorithm based on hierarchical sub-graph selection technique (HSGS) has been developed. This algorithm aims at overcoming the limitations of large datasets in the standard face recognition techniques. This algorithm creates small sub-graphs, selects the best matches from each sub-graph, and then dynamically creates next-level sub-graphs until a single group remains. The best match from this final group is then accepted as the rank 1 final result of the face recognition process. The study also investigated the best approach for creating sub-graphs by developing an objective function that can be used for detecting the best dissimilarity between groups at all levels. Detailed testing on large benchmark datasets indicates that the proposed method is highly efficient with a sub-graph size of approximately 50 nodes (images) for the Eigenface technique. Compared to the standard Eigenface algorithm, the new hierarchical sub-graph selection technique improved the recognition rate by more than 40% on some datasets, and by more than 2% compared to the mean based dissimilarity method. The future work will focus on applying the hierarchical technique to additional unsupervised face recognition algorithms such as LBP, SURF and other unsupervised computer vision algorithms that suffer from degradation in recognition rate due to large dataset sizes. This system also proposed integrating different preprocessing modules, such as super-resolution

and 3D alignment, that can be used to improve the quality of input query images in wild environments. A comprehensive study about the effect of these two modules on high dimensional features for unsupervised face recognition has been introduced. This study concluded that applying the super-resolution algorithm after the 3D alignment improves the recognition rate because the interpolation steps involved in the alignment module. The system also introduced the integration of face recognition and verification by using Siamese network on top 50 matches to improve rank 1 recognition rate of the overall framework. Further improvements can be achieved for this framework by adjusting the parameters of Siamese network, alignment module, super-resolution and the hierarchical sub-graph algorithms. The framework has been applied to challenging datasets such as FERET and FRGC v2.0 to test the robustness.

The second system introduced in this work is a 3D based framework for face recognition and verification. The proposed framework developed neural generative models from the input registered 3D point clouds, and used the generated models as a new features representation for the original clouds. Experiments on Bosporus dataset showed that the average Mean Square Error (MSE) for all generated models is 0.00037 (which is very low and close to the target MSE 0.0002) and the accuracy for representing the original point clouds is above 99%. The models resulted in improved noise reduction over the original cloud. An average scaling factor of 80 has been achieved between the size of the original stored point clouds and the generated models representations. The achieved verification rate of the generated models using Siamese Network with a binary classifier was over 98%. For the 3D based recognition framework, further improvement can be achieved by improving the registration step for the gallery and query input point clouds and more testings can be done on faces with different expressions and resolutions.

References

[1] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proceedings CVPR '91., IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991.*, pp. 586–591, Jun 1991.

[2] H. Moon and P. J. Phillips, "Computational and performance aspects of pca-based face-recognition algorithms," *Perception*, vol. 30, no. 3, pp. 303–321, 2001.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, pp. 71–86, Jan. 1991.

[4] W. Zhao, R. Chellappa, and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," in *Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998.*, pp. 336–341, Apr 1998.

[5] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using lda-based algorithms," *Trans. Neur. Netw.*, vol. 14, pp. 195–200, Jan. 2003.

[6] M. Bartlett, J. R. Movellan, and T. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks,*, vol. 13, pp. 1450–1464, Nov 2002.

[7] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, pp. 1–48, Mar. 2003.

[8] M.-H. Yang, "Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, FGR '02, (Washington, DC, USA), pp. 215–, IEEE Computer Society, 2002.

[9] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, pp. 1299–1319, July 1998.

[10] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, pp. 469–481, 2004.

[11] D. Maturana, D. Mery, and A. Soto, "Face recognition with local binary patterns, spatial pyramid histograms and naive bayes nearest neighbor classification," in *Proceedings of the 2009 International Conference of the Chilean Computer Science Society*, SCCC '09, (Washington, DC, USA), pp. 125–132, IEEE Computer Society, 2009.

[12] L. Zhang, J. Chen, Y. Lu, and P. Wang, "Face recognition using scale invariant feature transform and support vector machine," in *The 9th International Conference for Young Computer Scientists, 2008. ICYCS 2008.*, pp. 1766–1770, Nov 2008.

[13] C. Geng and X. Jiang, "Face recognition using sift features," in *Proceedings of the 16th IEEE International Conference on Image Processing*, ICIP'09, (Piscataway, NJ, USA), pp. 3277–3280, IEEE Press, 2009.

[14] S.-H. Tse and K.-M. Lam, "Efficient face recognition with a large database," in *ICARCV 2008. 10th International Conference on Control, Automation, Robotics and Vision, 2008.*, pp. 944–949, Dec 2008.

[15] J. Lu and K. Plataniotis, "Boosting face recognition on a large-scale database," in *Proceedings. 2002 International Conference on Image Processing, 2002.*, vol. 2, pp. II–109–II–112 vol.2, 2002.

[16] M. Kyperountas, A. Tefas, and I. Pitas, "Face recognition via adaptive discriminant clustering," in *15th IEEE International Conference on Image Processing, 2008. ICIP 2008.*, pp. 2744–2747, Oct 2008.

[17] J. Lu, K. Plataniotis, and A. Venetsanopoulos, "Boosting linear discriminant analysis for face recognition," in *Proceedings. 2003 International Conference on Image Processing, ICIP 2003.*, vol. 1, pp. I–657–60 vol.1, Sept 2003.

[18] L. Best, H. Han, C. Otto, B. Klare, and A. K. Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," Tech. Rep. MSU-CSE-14-1, Department of Computer Science, Michigan State University, East Lansing, Michigan, March 2014.

[19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, (Washington, DC, USA), pp. 1701–1708, IEEE Computer Society, 2014.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, (Washington, DC, USA), pp. 1891–1898, IEEE Computer Society, 2014.

[22] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *CoRR*, vol. abs/1406.4773, 2014.

[23] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *2013 IEEE International Conference on Computer Vision (ICCV),*, pp. 1489–1496, Dec 2013.

[24] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *CoRR*, vol. abs/1502.00873, 2015.

[25] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with gaussianface," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 3811–3819, 2015.

[26] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013*, pp. 3025–3032, June 2013.

[27] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. V. Rohith, "Fully automatic pose-invariant face recognition via 3d pose normalization," in *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pp. 937–944, 2011.

[28] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov 1998.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR 2015*, 2015.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Neurocomputing: Foundations of research," ch. Learning Representations by Back-propagating Errors, pp. 696–699, Cambridge, MA, USA: MIT Press, 1988.

[33] M. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks,*, vol. 5, pp. 989–993, Nov 1994.

[34] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*, pp. 184–199, 2014.

[35] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[36] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, vol. 1, pp. 539–546 vol. 1, June 2005.

[37] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[38] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," Tech. Rep. UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.

[39] F. Lin, C. Fookes, V. Chandran, and S. Sridharan, *Super-Resolved Faces for Improved Face Recognition from Surveillance Video*, pp. 1–10. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[40] F. W. Wheeler, X. Liu, and P. H. Tu, "Multi-frame super-resolution for face recognition," in *2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pp. 1–6, Sept 2007.

[41] S. Hu, R. Maschal, S. S. Young, T. H. Hong, and P. J. Phillips, "Face recognition performance with superresolution," *Appl. Opt.*, vol. 51, pp. 4250–4259, Jun 2012.

[42] Y. Kong, S. Zhang, and P. Cheng, "Super-resolution reconstruction face recognition based on multi-level {FFD} registration," *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 24, pp. 6926 – 6931, 2013.

[43] C. Fookes, F. Lin, V. Chandran, and S. Sridharan, "Evaluation of image resolution and super-resolution on face recognition performance," *Journal of Visual Communication and Image Representation*, vol. 23, no. 1, pp. 75 – 93, 2012.

[44] P. Rasti, T. Uiboupin, S. Escalera, and G. Anbarjafari, *Convolutional Neural Network Super Resolution for Face Recognition in Surveillance Monitoring*, pp. 175–184. Cham: Springer International Publishing, 2016.

[45] A. ElSayed, A. Mahmood, and T. Sobh, *Unsupervised Sub-graph Selection and Its Application in Face Recognition Techniques*, pp. 247–256. Cham: Springer International Publishing, 2015.

[46] P. Dreuw, P. Steingrube, H. Hanselmann, and H. Ney, "Surf-face: Face recognition under viewpoint consistency constraints," in *Proc. BMVC*, pp. 7.1–7.11, 2009. doi:10.5244/C.23.7.

[47] S. Liao, Z. Lei, D. Yi, and S. Z. Li, "A benchmark study of large-scale unconstrained face recognition," in *IEEE International Joint Conference on Biometrics*, pp. 1–8, Sept 2014.

[48] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, June 2005.

[49] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, (Washington, DC, USA), pp. 1867–1874, IEEE Computer Society, 2014.

[50] M. Abualkibash, A. ElSayed, and A. Mahmood, "Highly scalable, parallel and distributed adaboost algorithm using light weight threads and web services on a network of multi-core machines," *CoRR*, vol. abs/1306.1467, 2013.

[51] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–511–I–518 vol.1, 2001.

[52] "Extended yale b+ dataset,"

[53] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 22, pp. 1090–1104, Oct 2000.

[54] S. Rizvi, P. Phillips, and H. Moon, "The feret verification testing protocol for face recognition algorithms," in *Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998.*, pp. 48–53, April 1998.

[55] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, (Washington, DC, USA), pp. 947–954, IEEE Computer Society, 2005.

[56] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge, "Recognizing faces with PCA and ICA," *Computer Vision and Image Understanding*, vol. 91, no. 1-2, pp. 115–137, 2003.

[57] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 32, pp. 2106–2112, Nov 2010.

[58] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 23, pp. 643–660, Jun 2001.

[59] Y. Tang, R. Salakhutdinov, and G. Hinton, "Robust boltzmann machines for recognition and denoising," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*, pp. 2264–2271, June 2012.

[60] T. Ahonen, S. Member, A. Hadid, M. Pietikäinen, and S. Member, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2037–2041, 2006.

[61] L. Wiskott, N. Krüger, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 775–779, July 1997.

[62] B. Moghaddam, C. Nastar, and A. Pentland, "Bayesian face recognition using deformable intensity surfaces," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 638–645, June 1996.

[63] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, *Learning Multi-scale Block Local Binary Patterns for Face Recognition*, pp. 828–837. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[64] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, *Boosting Local Binary Pattern (LBP)-Based Face Recognition*, pp. 179–186. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[65] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition," *Comput. Vis. Image Underst.*, vol. 101, pp. 1–15, Jan. 2006.

[66] A. Scheenstra, A. Ruifrok, and R. Veltkamp, "A survey of 3d face recognition methods," in *Audio- and Video-Based Biometric Person Authentication* (T. Kanade, A. Jain, and N. Ratha, eds.), vol. 3546 of *Lecture Notes in Computer Science*, pp. 891–899, Springer Berlin Heidelberg, 2005.

[67] M. Daoudi, A. Srivastava, and R. Veltkamp, *3D Face Modeling, Analysis and Recognition*. Wiley Publishing, 1st ed., 2013.

[68] J. Y. Cartoux, J. T. LaPreste, and M. Richetin, "Face authentication or recognition by profile extraction from range images," in *Proceedings of the Workshop on Interpretation of 3D Scenes*, pp. 194–199, November 1989.

[69] J. Lee and E. Milios, "Matching range images of human faces," in *Proceedings, Third International Conference on Computer Vision, 1990.*, pp. 722–726, Dec 1990.

[70] G. Gordon, "Face recognition based on depth and curvature features," in *Proceedings CVPR '92., 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1992.*, pp. 808–810, Jun 1992.

[71] T. Nagamine, T. Uemura, and I. Masuda, "3d facial image analysis for human identification," in *Proceedings., 11th IAPR International Conference on Pattern*

Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications,, pp. 324–327, Aug 1992.

[72] B. Achermann, X. Jiang, and H. Bunke, "Face recognition using range images," in *Proceedings., International Conference on Virtual Systems and MultiMedia, 1997. VSMM '97.*, pp. 129–136, Sep 1997.

[73] H. Tanaka, M. Ikeda, and H. Chiaki, "Curvature-based face surface recognition using spherical correlation. principal directions for curved object recognition," in *Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998.*, pp. 372–377, Apr 1998.

[74] B. Achermann and H. Bunke, "Classifying range images of human faces with hausdorff distance," in *2000. Proceedings. 15th International Conference on Pattern Recognition,*, vol. 2, pp. 809–813 vol.2, 2000.

[75] C. Hesher, A. Srivastava, and G. Erlebacher, "A novel technique for face recognition using range imaging," in *Proceedings. Seventh International Symposium on Signal Processing and Its Applications, 2003.*, vol. 2, pp. 201–204 vol.2, July 2003.

[76] J. Min, K. W. Bowyer, and P. Flynn, "Using multiple gallery and probe images per person to improve performance of face recognition," technical report, Notre Dame Computer Science and Engineering Technical Report, 2003.

[77] G. Medioni and R. Waupotitsch, "Face recognition and modeling in 3d," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2003)*, pp. 232–233, October 2003.

[78] A. B. Moreno, Ángel Sánchez, J. F. Vélez, and F. J. Díaz, "Face recognition using 3d surface-extracted descriptors," in *In Irish Machine Vision and Image Processing Conference (IMVIP 2003), Sepetember*, 2003.

[79] Y. Lee, K. Park, J. Shim, and T. Yi, "3d face recognition using statistical multiple features for the local depth information," in *16th International Conference on Vision Interface*, June 2003.

[80] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Expression-invariant 3d face recognition.," in *AVBPA* (J. Kittler and M. S. Nixon, eds.), vol. 2688 of *Lecture Notes in Computer Science*, pp. 62–69, Springer, 2003.

[81] N. Alyuz, B. Gokberk, and L. Akarun, "A 3d face recognition system for expression and occlusion invariance," in *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*, pp. 1–7, Sept 2008.

[82] H. Li, D. Huang, P. Lemaire, J. M. Morvan, and L. Chen, "Expression robust 3d face recognition via mesh-based histograms of multiple order surface differential quantities," in *2011 18th IEEE International Conference on Image Processing*, pp. 3053–3056, Sept 2011.

[83] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen, "Feature detection on 3d face surfaces for pose normalisation and recognition," in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6, Sept 2010.

[84] R. Min, J. Choi, G. Medioni, and J. Dugelay, "Real-time 3d face identification from a depth camera," in *2012 21st International Conference on Pattern Recognition (ICPR),*, pp. 1739–1742, Nov 2012.

[85] A.-M. Cretu, E. Petriu, and G. Patry, "Neural-network-based models of 3-d objects for virtualized reality: a comparative study," *IEEE Transactions on Instrumentation and Measurement,*, vol. 55, pp. 99–111, Feb 2006.

[86] Richard Socher and Brody Huval and Bharath Bhat and Christopher D. Manning and Andrew Y. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification," in *Advances in Neural Information Processing Systems 25*, 2012.

[87] L. Alexandre, "3d object recognition using convolutional neural networks with transfer learning between input channels," in *Intelligent Autonomous Systems 13* (E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, eds.), vol. 302 of *Advances in Intelligent Systems and Computing*, pp. 889–898, Springer International Publishing, 2016.

[88] V. Nair and G. E. Hinton, "3d object recognition with deep belief nets," in *Advances in Neural Information Processing Systems 22* (Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, eds.), pp. 1339–1347, Curran Associates, Inc., 2009.

[89] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, Feb. 1992.

[90] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

[91] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Biometrics and identity management," ch. Bosphorus Database for 3D Face Analysis, pp. 47–56, Berlin, Heidelberg: Springer-Verlag, 2008.

[92] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 711–720, July 1997.

[93] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," *Journal of Optical Society of America A*, vol. 14, pp. 1724–1733, 1997.

[94] C. Liu and H. Wechsler, "Comparative assessment of independent component analysis (ica) for face recognition," in *International Conference on Audio and Video Based Biometric Person Authentication*, pp. 22–24, 1999.

[95] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 228–233, Feb. 2001.

[96] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proceedings CVPR '94., 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994.*, pp. 84–91, Jun 1994.